What is Version Control System (VCS)?

A **Version Control System (VCS)** is a tool used to track and manage changes to files over time. It helps developers:

- Save different versions of code, documents, or other files.
- Collaborate on projects without overwriting each other's work.
- Revert to previous versions in case of errors or bugs.

There are two main types of VCS:

1. **Centralized VCS (CVCS)**: A single server stores all versions, and users pull changes from it (e.g., SVN, Perforce).
   Limitation: If the server is down, collaboration halts.
2. **Distributed VCS (DVCS)**: Every user has a full copy of the repository, allowing local commits and operations (e.g., Git, Mercurial).
   Advantage: Allows offline work and greater redundancy.

Git, a distributed VCS, is the most widely used system because of its speed, reliability, and features like branching and merging.

---

2. Learn Version Control (Git)

Basics of Git

1. **What is Git?**
    - Git is a **distributed version control system** that allows multiple developers to work on a project simultaneously without overwriting each other's changes. Each developer has their own local copy of the entire repository, and changes can be synced with the remote repository (e.g., on GitHub or GitLab).

2. **Installing Git**:
   - To install Git, follow the instructions based on your operating system:
     - **Windows**: Download the installer from Git for Windows and follow the setup instructions.
     - **Mac**: Use Homebrew: brew install git or download the installer from Git's official site.
     - **Linux**: Use your package manager, e.g., sudo apt-get install git for Ubuntu or sudo yum install git for Fedora.
3. **Configure Git**: After installation, configure your Git settings (name and email) so your commits are associated with your identity.

```
git config --global user.name "Your Name"
git config --global user.email "youremail@example.com"
```

Initializing a Git Repository:

- To start using Git in a project, navigate to the project directory and run:

```
git init
```

This will create a .git directory, turning your folder into a Git repository.

Cloning a Repository:

- If you want to work on an existing project, you can clone a repository from GitHub or GitLab:

```
git clone https://github.com/username/repository.git
```

This copies the entire project (including the history) to your local machine.

Core Git Commands

git add:

The git add command adds changes in your working directory to the staging area. The staging area is where changes are prepared before committing.

```
git add file.txt       # Add a specific file
git add .              # Add all files (recursively)
```

git commit:

- The git commit command records changes made to the repository. A commit is a snapshot of your code at a specific point in time.

```
git commit -m "Commit message"  # Commit with a message
```

git push:

- The git push command uploads your local commits to a remote repository, such as GitHub or GitLab.

```
git push origin main  # Push changes to the 'main' branch
```

origin refers to the remote repository, and main refers to the branch name.

git pull:

- The git pull command fetches changes from the remote repository and integrates them with your local branch.

```
git pull origin main  # Pull changes from the 'main' branch
```

git merge:

- The git merge command integrates changes from different branches. For example, if you want to merge changes from feature-branch into main, you would:

```
git checkout main            # Switch to the main branch
git merge feature-branch     # Merge the feature branch into main
```

This can result in conflicts if changes are made to the same part of the code, which you must resolve manually.

`git branch`:

- The `git branch` command is used to manage branches in Git. Branches allow you to work on different features or fixes independently.

```
git branch                   # List all branches
git branch new-feature       # Create a new branch
git checkout new-feature     # Switch to the new branch
```

GitHub

What is GitHub?

- GitHub is a **cloud-based platform** that provides hosting for Git repositories.
- It allows developers to store, manage, and collaborate on code projects.
- GitHub is widely used for open-source projects and provides tools for version control, issue tracking, and collaboration.

Parent Company:

- GitHub is owned by **Microsoft** (acquired in 2018).

Key Features of GitHub:

1. **Repositories**:
    - Store project code and files in repositories (public or private).
    - Supports unlimited collaborators.
2. **Pull Requests**:
    - A way to propose changes to a repository.
    - Enables team members to review and discuss code changes before merging.
3. **Issues**:
    - Used to track bugs, tasks, or feature requests.
    - Assign labels, milestones, and team members for better organization.
4. **GitHub Actions**:
    - Built-in Continuous Integration/Continuous Deployment (CI/CD) tool.

- Automates workflows, such as testing and deployment.

5. **GitHub Pages**:
    - Hosts static websites directly from a GitHub repository.
    - Supports custom domains and built-in themes.
6. **Community Features**:
    - Allows developers to fork repositories, watch projects, and star repositories.
    - Provides README files and markdown support for documentation.
7. **Security Tools**:
    - Dependabot for dependency vulnerability scanning.
    - Security advisories and secret scanning.

Collaboration Workflow on GitHub:

1. Clone the repository locally.
2. Create a new branch for your changes.
3. Commit and push changes to your branch.
4. Open a pull request for review.
5. Address feedback, if any, and merge the pull request.

Pricing:

- Free tier for public and private repositories.
- Paid plans for advanced features like more storage, enterprise options, and advanced security tools.

GitLab

What is GitLab?

- GitLab is an **open-source DevOps platform** that provides Git repository hosting.
- It offers integrated CI/CD pipelines, issue tracking, and tools to automate the software development lifecycle (SDLC).
- It can be used as a cloud-hosted service or installed on-premises.

Parent Company:

- GitLab is owned by **GitLab Inc.**, an independent company.

Key Features of GitLab:

1. **Repositories**:
    - Store code in repositories with support for branching, merging, and file history.
    - Provides visibility into repository activity and contributions.
2. **Merge Requests**:
    - GitLab's equivalent of GitHub's pull requests.
    - Includes inline code reviews, discussions, and pipelines.
3. **CI/CD Pipelines**:
    - Fully integrated CI/CD tools for automated testing, building, and deployment.
    - Customizable pipeline configurations using .gitlab-ci.yml file.

4. **Project Management**:
    - Features like boards, milestones, and issues to manage development tasks.
    - Time tracking for better productivity analysis.
5. **DevSecOps**:
    - Security scanning integrated into pipelines.
    - Tools for static application security testing (SAST), dependency scanning, and more.
6. **Kubernetes Integration**:
    - Simplifies deploying applications to Kubernetes clusters.
    - Offers features for monitoring and managing Kubernetes environments.
7. **Self-Hosting**:
    - GitLab can be hosted on your own servers for full control and customization.
    - Provides enterprise-grade features for larger organizations.

Collaboration Workflow on GitLab:

1. Clone the repository locally.
2. Create a new branch and commit changes.
3. Push changes to the branch and open a merge request.
4. Review, resolve conflicts if any, and merge the request.

Pricing:

- Free tier for small teams and personal use.
- Paid plans include premium DevOps features and enterprise support.

Comparison: Git vs GitHub vs Bitbucket

Feature	Git	GitHub	Bitbucket
Definition	Distributed version control system.	Git repository hosting platform.	Git repository hosting platform.
Purpose	Tracks changes in files and manages repositories.	Collaboration and hosting for Git repositories.	Collaboration and hosting for Git repositories.
Hosting	Does not include hosting.	Cloud-hosted.	Cloud-hosted and self-hosted.
Repositories	Local repositories.	Unlimited public/private repos (Free plan).	Unlimited private repos (Free plan).
CI/CD Integration	None.	GitHub Actions (built-in).	Bitbucket Pipelines (built-in).
Security Features	None.	Dependabot, secret scanning.	Integrated security scanning.
Self-Hosting	Not applicable.	Not supported.	Supported with Bitbucket Server.
Pricing	Free.	Free and paid plans.	Free and paid plans.
Popular Use Cases	Local version control.	Open-source projects, collaboration.	Enterprise-level private projects.
Community Support	Vast (used with GitHub and others).	Huge community for open-source projects.	Smaller community compared to GitHub.
Kubernetes Support	None.	Limited.	Better Kubernetes integration.