

Git

Git is a **distributed version control system** (VCS) that allows multiple developers to work on the same project simultaneously. It tracks changes to files, enabling efficient collaboration, managing code versions, and ensuring project history is well-organized. Git is widely used in software development for managing source code, and it enables teams to keep track of changes, revert to previous versions, and work on branches without affecting the main project.

Key Features of Git:

1. Distributed Version Control

- a. Git allows every developer to have a complete copy of the repository, including its entire history, on their local machine. This ensures better redundancy and faster operations.

2. Branching and Merging

- a. Developers can create isolated branches to work on features or fixes without affecting the main codebase. Git makes it easy to merge changes back into the main branch once the work is completed.

3. Version History

- a. Every change made to the project is recorded, allowing you to track who made changes, what was changed, and why. This enables easy rollback and better debugging.

4. Collaboration

- a. Git makes collaboration among developers easier, enabling them to work on the same project concurrently. It also allows for managing code from different sources and ensuring that changes can be integrated.

5. Open Source

- a. Git is open-source software, meaning it is free to use and can be modified. It is supported by a large community of developers, ensuring continuous improvements and support.

Core Git Concepts:

1. Repository (Repo):

2. A repository is where your project files and their version history are stored. It can be local (on your computer) or remote (on a server like GitHub or GitLab).

3. Commit:

A commit is a snapshot of your project at a specific point in time. It includes a unique identifier, a timestamp, and a message explaining the changes made.

4. Branch:

A branch is a separate line of development. The default branch in Git is usually called **main** (or **master**). Developers can create new branches to work on features or bug fixes.

5. Merge:

Merging combines the changes from different branches. After work on a branch is completed, you can merge it back into the main branch or another branch.

6. Pull:

A pull is a command used to fetch changes from a remote repository and automatically merge them into your local copy of the repository.

7. Push:

Pushing sends your local commits to a remote repository, making them available to other developers working on the project.

Basic Git Commands:

1. git init:

2. Initializes a new Git repository in your current directory.

```
C:\Users\Sandhya Rani\Documents\Github\OOPS>echo "# OOPS-Concepts-in-CPP" >> README.md
C:\Users\Sandhya Rani\Documents\Github\OOPS>git init
Initialized empty Git repository in C:/Users/Sandhya Rani/Documents/Github/OOPS/.git/
C:\Users\Sandhya Rani\Documents\Github\OOPS>git add README.md
```

3. **git clone [repository URL]:**

Creates a copy of a remote repository on your local machine.

```
C:\Users\Sandhya Rani\Documents\Github\OOPS>git remote add origin https://github.com/Sandhya990/OOPS-Concepts-in-CPP.git
```

4. **git status:**

Displays the current state of your working directory and staging area (changes that are staged, unstaged, or untracked).

```
C:\Users\Sandhya Rani\Documents\Github\OOPS>git branch -M main
```

5. **git add [file]:**

Adds changes to the staging area, preparing them to be committed.

6. **git commit -m "message":**

Commits the staged changes to the local repository with a descriptive message.

```
C:\Users\Sandhya Rani\Documents\Github\OOPS>git commit -m "first commit"
[master (root-commit) 7660d8f] first commit
1 file changed, 1 insertion(+)
create mode 100644 README.md
```

7. **git push:**

Pushes your local commits to a remote repository.

```
C:\Users\Sandhya Rani\Documents\Github\OOPS>git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 244 bytes | 244.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Sandhya990/OOPS-Concepts-in-CPP.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

8. **git pull:**

Pulls changes from the remote repository and merges them with your local repository.

9. git branch [branch name]:

Creates a new branch.

10. git checkout [branch name]:

Switches to a different branch.

11. git merge [branch name]:

Merges changes from one branch into the current branch.

Advantages of Using Git:

1. Speed and Efficiency:

2. Git operations (e.g., committing, branching, merging) are fast, even with large repositories.

3. Collaboration and Code Sharing:

Git makes it easy to share code and collaborate with others, even across different teams and locations.

4. Branching and Parallel Development:

Multiple developers can work on different parts of the project simultaneously without interfering with each other's work.

5. History and Auditability:

Git's version control system records a detailed history of changes, which can be helpful for auditing and troubleshooting.

6. Supports Large Projects:

Git can handle large codebases efficiently, even with a high volume of changes.

Common Git Platforms:

- **GitHub:** A popular platform that hosts Git repositories and provides tools for collaboration, issue tracking, pull requests, and code reviews.
- **GitLab:** A web-based Git repository manager that offers features like CI/CD (Continuous Integration and Delivery), issue tracking, and project management.
- **Bitbucket:** Another Git-based platform primarily used for private repositories and integrating with Atlassian tools like Jira.

When to Use Git:

- **Collaborative Projects:** When multiple people are working on the same codebase.
- **Open-Source Projects:** To manage contributions and track changes.
- **Version Control:** When you need to keep track of changes over time and be able to roll back if needed.
- **Distributed Workflows:** If team members work in different locations, Git allows seamless integration of their changes.