# Git And Version Control

**Version Control:** A system that tracks changes in files over time, allowing you to revert, collaborate, and manage versions.

**Git:** A distributed version control system that runs on your local machine to track file changes and manage code history.

**GitHub:** A cloud-based Git repository hosting service that enables collaboration, pull requests, and remote storage of Git repositories.

**Git Architecture: Working Directory, Staging Area, Local Repository, Remote Repository**

Git organizes files and changes into four key areas:

**1. Working Directory (Working Tree)**

* This is where you make changes to files in your project.
* Any modifications to files start here.
* Untracked and modified files exist in this area.
* Example: When you open a project folder and edit a file, it’s in the working directory.

**2. Staging Area (Index)**

* This is an intermediate area where Git tracks changes before committing them.
* Files must be added to the staging area before committing.
* Example: You edit a file but don’t want to commit all changes yet.

**3. Local Repository**

* This is where committed changes are stored on your computer.
* Every commit in the local repository is tracked with a unique SHA-1 hash.
* Example: Your commit history exists in your local Git folder (.git).

**4. Remote Repository**

* This is a repository hosted on a server like **GitHub, GitLab, Bitbucket, or Azure DevOps**.
* It allows collaboration with other developers.
* Changes must be pushed from the local repository to the remote repository.

**Git Workflow Example: Open, Modify, Add, Commit, and Push**

**Step 1: Open and Navigate to Your Repository**

If you already have a repository, navigate to it in the terminal:

cd path/to/your-repository

If you don’t have a Git repository, create one:

mkdir my-project

cd my-project

git init # Initialize a new Git repository

**Step 2: Create and Open a File**

Create a new file using touch:

touch chapter1.txt

nano chapter1.txt # For Linux/macOS

notepad chapter1.txt # For Windows

code chapter1.txt # If using VS Code

Add some text inside the file, save it, and close.

**Step 3: Check File Status**

To see which files have been modified:

git status

You should see chapter1.txt as an untracked file.

**Step 4: Add the File to the Staging Area**

Move the file to the **staging area**:

git add chapter1.txt

Check the status again:

git status

Now, the file is in the staging area (ready for commit).

**Step 5: Commit the File to the Local Repository**

Save the changes in the **local repository**:

git commit -m "Added chapter1.txt"

Now, the file is tracked in your local repository.

**Step 6: Connect to a Remote Repository (GitHub Example)**

If you haven't linked a remote repository yet:

1. Go to GitHub and create a new repository (e.g., my-project).
2. Copy the repository URL.
3. Add it as a remote repository:

git remote add origin https://github.com/your-username/my-project.git

To check if the remote is set up:

git remote -v

**Step 7: Push the Commit to the Remote Repository**

To upload the changes to GitHub:

git push origin main # If your branch is "main"

If it’s your first push, you may need to set the upstream branch:

git push -u origin main