Git & GitHub

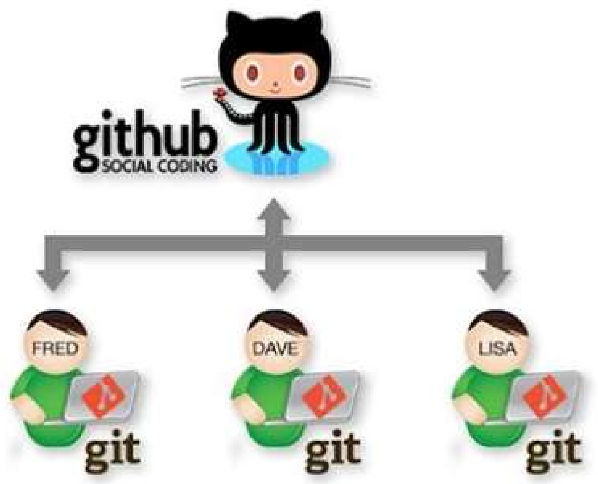
Git

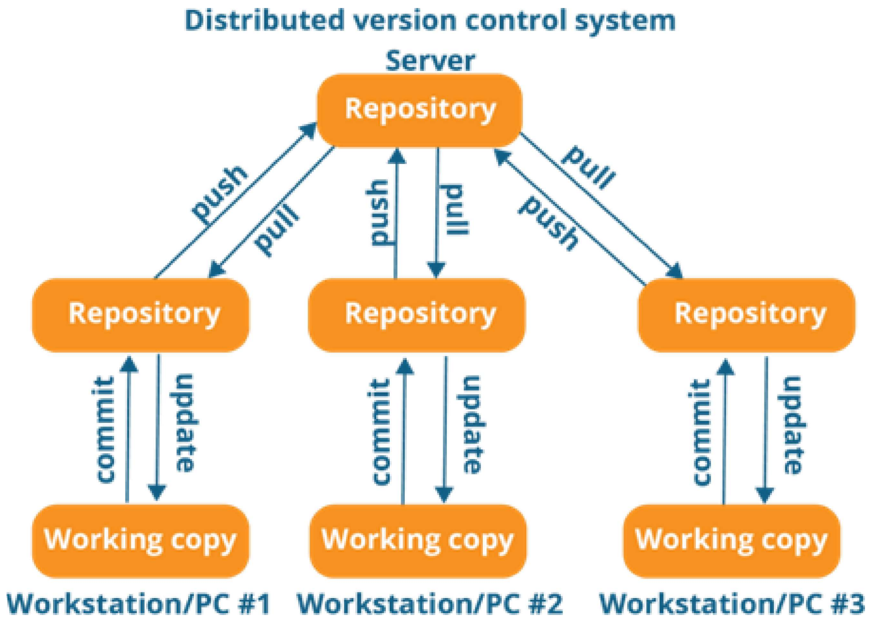
- ➔ Git is a distributed version control system that helps developers track changes in code and collaborate on projects.

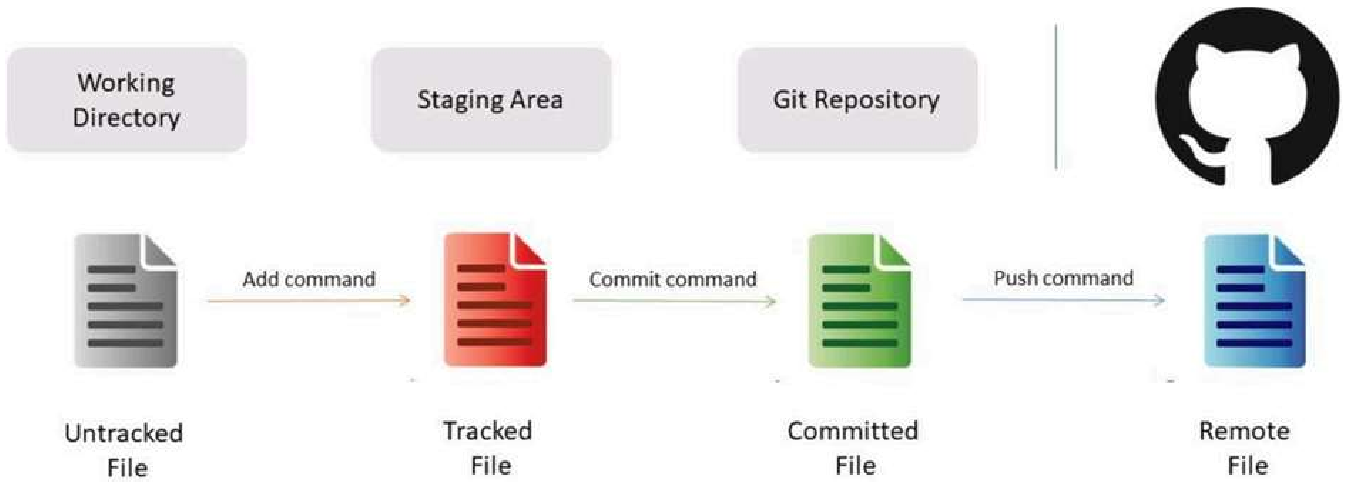
GitHub

- ➔ GitHub is a cloud-based platform for hosting Git repositories, enabling collaboration and version control.

Git & GitHub Workflow







Steps for Git workflow

1. Initialize a repository.
2. Configure user information.
3. Stage files for a commit.
4. Commit changes locally.
5. Link to a remote repository.
6. Push changes to the remote repository.
7. Pull latest changed from remote repository.
8. Delete local repository (Optional)

Create a New Local Git Repository

- ➔ To initialize a new Git repository in a directory, navigate to the desired folder and execute the following command
  - ◆ git init

Configure User Information (One-Time Setup)

- ➔ Set your username and email for Git. This is typically a one-time setup for identifying the author of the commits. Replace "your name" and "your email" with your actual details
  - ◆ git config --global user.name "your name"
  - ◆ git config --global user.email "your email"

Add Files or Folders to the Staging Area

- ➔ To stage changes for the next commit, use the git add command. You can stage specific files, all files, or files matching certain patterns
  - ◆ **Add all files and folders to staging:**
    - git add -A
  - ◆ **Add a specific file:**
    - git add filename
  - ◆ **Add all files with a specific extension (e.g., Java files):**
    - git add *.java
  - ◆ **Add all files within a folder:**
    - git add foldername
    - git add .

Commit Changes to the Local Repository

- ➔ After staging the files, commit them to your local repository with a descriptive commit message
  - ◆ git commit -m "commit message"

Connect Local Repository to a Remote Repository (One-Time Setup)

- ➔ Link your local repository to a remote repository using the git remote add command.
- ➔ Replace the example URL with the actual remote repository URL
  - ◆ git remote add origin "https://github.com/Madhan-091296/myproject.git"

Push Changes to the Remote Repository

- ➔ To upload your committed changes to the remote repository, use the git push command. Specify the remote name (origin) and the branch name (master for the main branch):
    - ◆ git push origin master
- ➔ Need to pass token which is generated in GitHub.

Pull the latest Changes from the Remote Repository

- ➔ To get latest changes from the remote repository to local repository , you need to use git pull command.
    - ◆ git fetch origin master

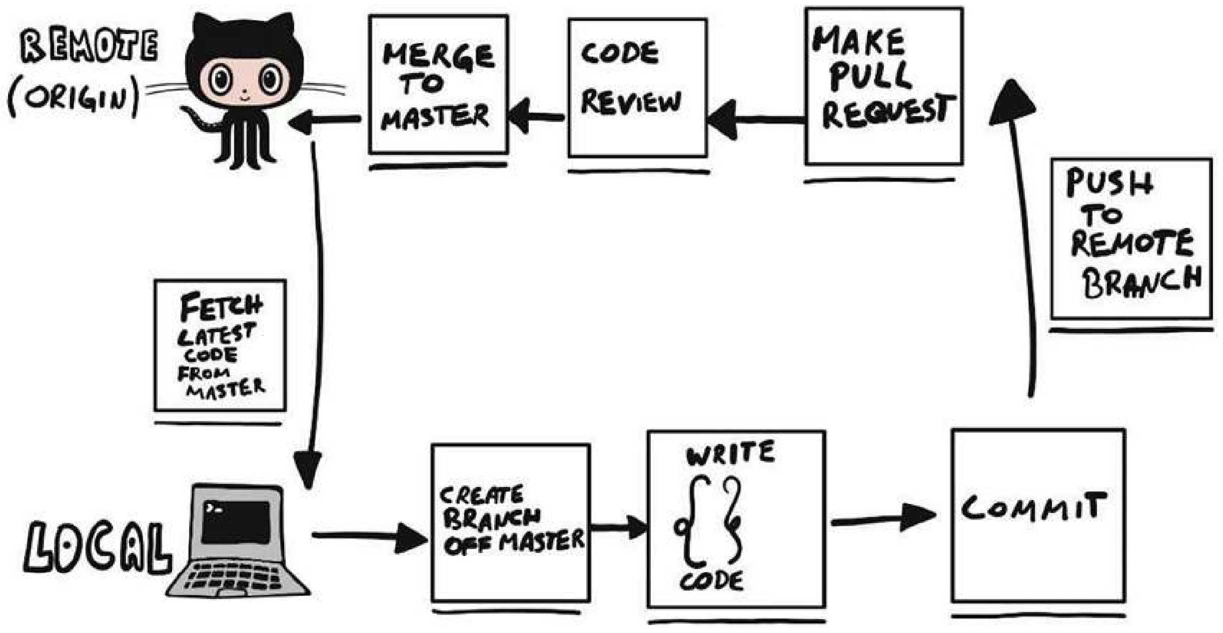
(or)

- ◆ git pull origin master

To Delete local repository

- ➔ rm -rf

Branching & Merging



Step 1: Fetch the latest code from master (remote)

- ➔ git fetch origin master

(or)

- ➔ git pull origin master

(or)

- ➔ git clone <<URL>>
- ➔ **Example:**
    - ◆ git clone "https://github.com/Madhan-091296/myproject.git"
- ➔ This updates your local master branch with the latest code from the remote repository.

Step 2: Create a new branch of master and switch to it.

- ➔ git branch <branch name>   // creates a new branch
- ➔ git checkout <branch name>   // switch to branch

- ➔ git checkout -b <branch-name>   // single command
- ➔ **Example:**
  - ◆ git branch Branch1
  - ◆ git checkout Branch1
  - ◆ git checkout -b Branch1   //single command for create a new branch and switch to it
- ➔ This creates and switches to a new branch called **Branch1** based on the master branch.

Step 3: Write code and make changes

- ➔ Make your changes to the files in your local repository. For example, you might edit file1.txt and create new files.

Step 4: Commit the changes

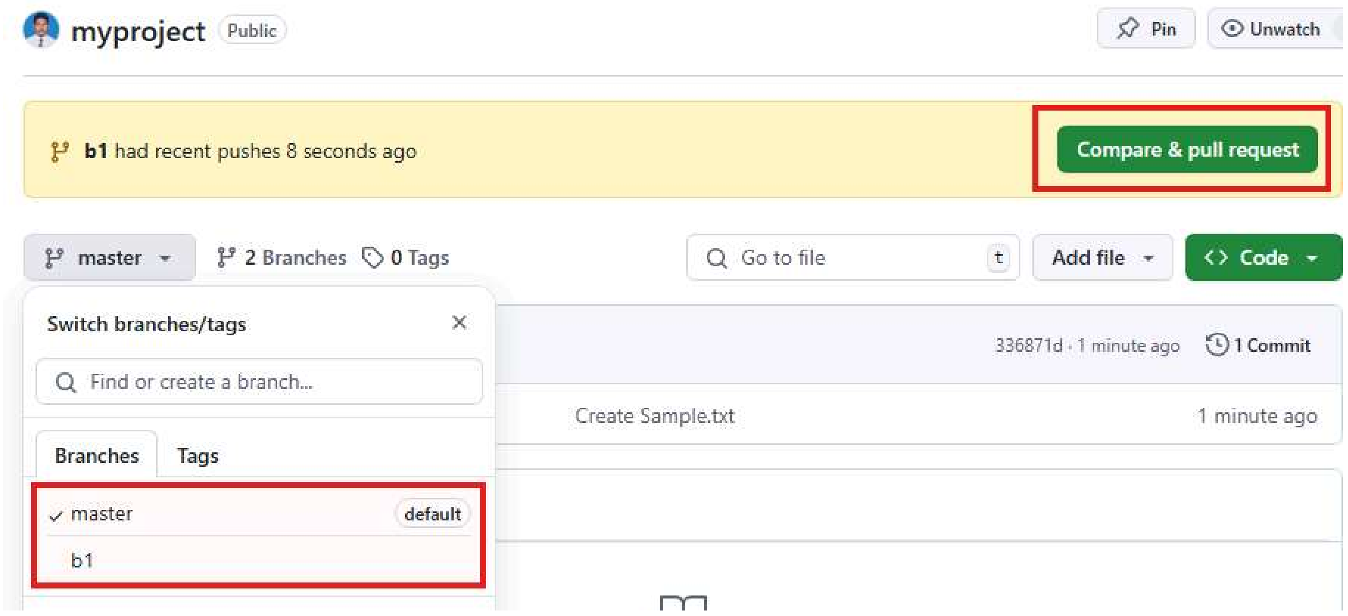
- ➔ git add .
- ➔ git commit -m "<commit-message>"
- ➔ **Example:**
  - ◆ git add .
  - ◆ git commit -m "Added file2.txt & modified file1.txt"
- ➔ This stages and commits all the changes in your branch.

Step 5: Push changes to the remote branch

- ➔ git push origin <new-branch-name>
- ➔ **Example:**
  - ◆ git push origin Branch1
- ➔ This uploads your branch to the remote repository.

Step 6: Make a pull request

- ➔ Go to your GitHub repository in a browser.
- ➔ You'll see a message like ***"Your branch has recent pushes"***.
- ➔ Click **Compare & Pull Request**.
- ➔ Add a title and description for the pull request, then submit it.


myproject Public

b1 had recent pushes 8 seconds ago

336871d · 1 minute ago 1 Commit

Create Sample.txt 1 minute ago

Step 7: Code review

- ➔ Collaborators review the pull request, add comments, and suggest changes.
- ➔ If there are changes requested, make them in your local branch, commit them, and push them again:
  - ◆ git commit -m "Updated file1.txt based on review feedback"
  - ◆ git push origin Branch1

Step 8: Merge to master

- ➔ After the code review is approved:
  - ◆ In GitHub, click **Merge Pull Request.**
  - ◆ Confirm the merge.
  - ◆ Your feature branch is now merged into master.

Optional Clean-Up: Delete the branch

- ➔ After merging, you can delete the branch locally and remotely.

Delete local branch:

- ➔ git branch -D <new-branch-name>

Delete remote branch:

- ➔ git push origin --delete <new-branch-name>
- ➔ **Example:**
  - ◆ git branch -D Branch1
  - ◆ git push origin --delete Branch1