Git & GitHub

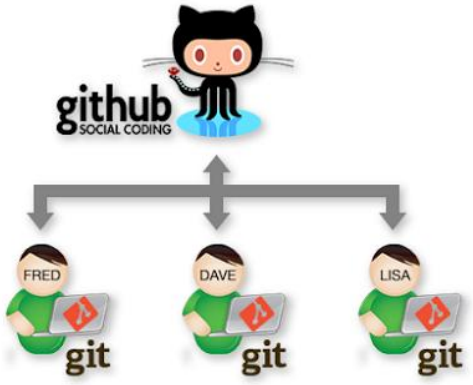
Git

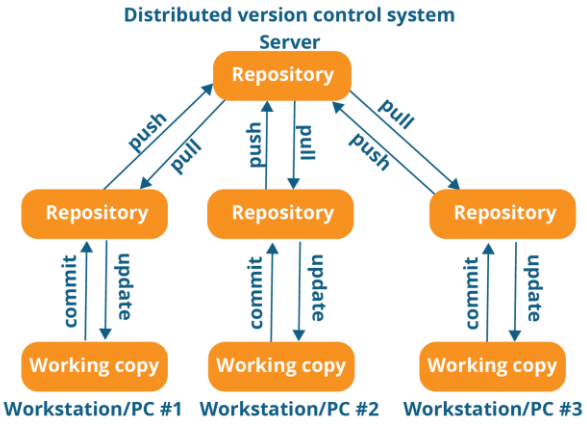
Git is a distributed version control system that helps developers track changes in code and collaborate on projects.

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)**

1. 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
```

2. 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"
```

3. 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 .
```

4. 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"
```

5. 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/pavanoltraining/myproject.git"
```

6. 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.

7. 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.

Option 1: Fetch and merge manually

```
git fetch origin master
```

```
git merge origin master
```

(Or)

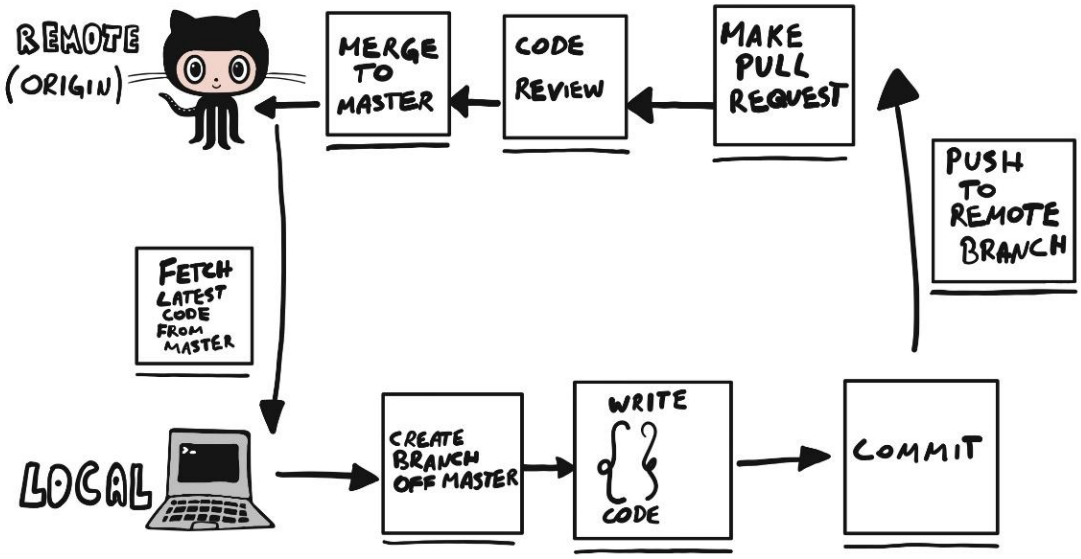
Option 2: Fetch and merge automatically

```
git pull origin master
```

8. To Delete local repository

```
rm -rf
```

Branching & Merging



Step 1: Clone the Repository (Fetch the Latest Code from master)

```
git clone <repository-URL>
```
Example:

Example:

```
git clone "https://github.com/pavanoltraining/myproject.git"
```

This will create a local copy of the repository with the latest code from the remote master branch.

Step 2: Create and Switch to a New Branch

Option 1: Separate commands

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

Option 2: Single command

```
git checkout -b <branch-name>
```

Example:

```
git checkout -b mybranch  //single command for create a new branch and switch to it
```

This creates a new branch **mybranch** and switches to it.

To check the current branch:

```
git branch --show-current
```

Step 3: Make Changes to the Code

Modify existing files or create new ones in your local repository as needed.

Step 4: Stage and Commit the Changes

Use the following commands to add and commit your changes:

```
git add .
git commit -m "<commit-message>"
```

Example:

```
git add .
git commit -m "Added file2 and modified file1"
```

This stages and commits all the changes in your branch.

Step 5: Push the Changes to the Remote Branch

```
git push origin <new-branch-name>
```

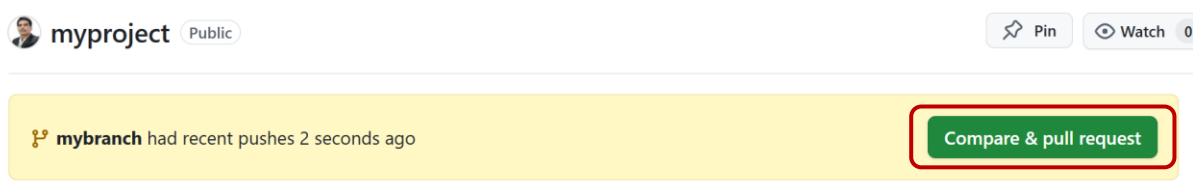
Example:

```
git push origin mybranch
```

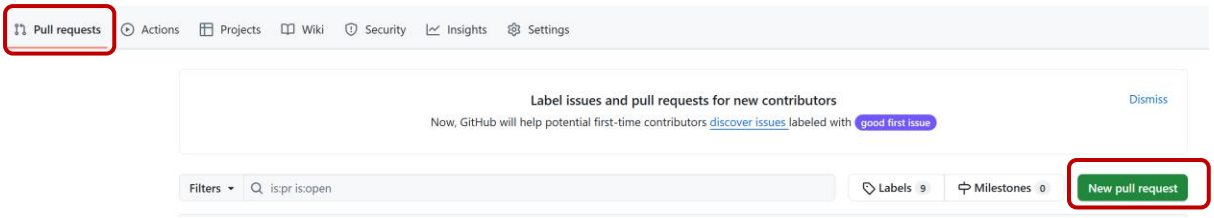
This uploads your branch to the remote repository.

Step 6: Create a Pull Request (PR)

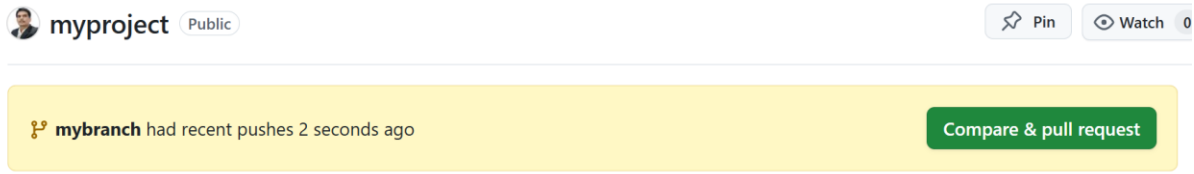
1. Open your GitHub repository in a browser.
2. GitHub will prompt you with a “Compare & pull request” option. (If there is no prompt) Navigate to **Pull requests** tab and create New pull request.

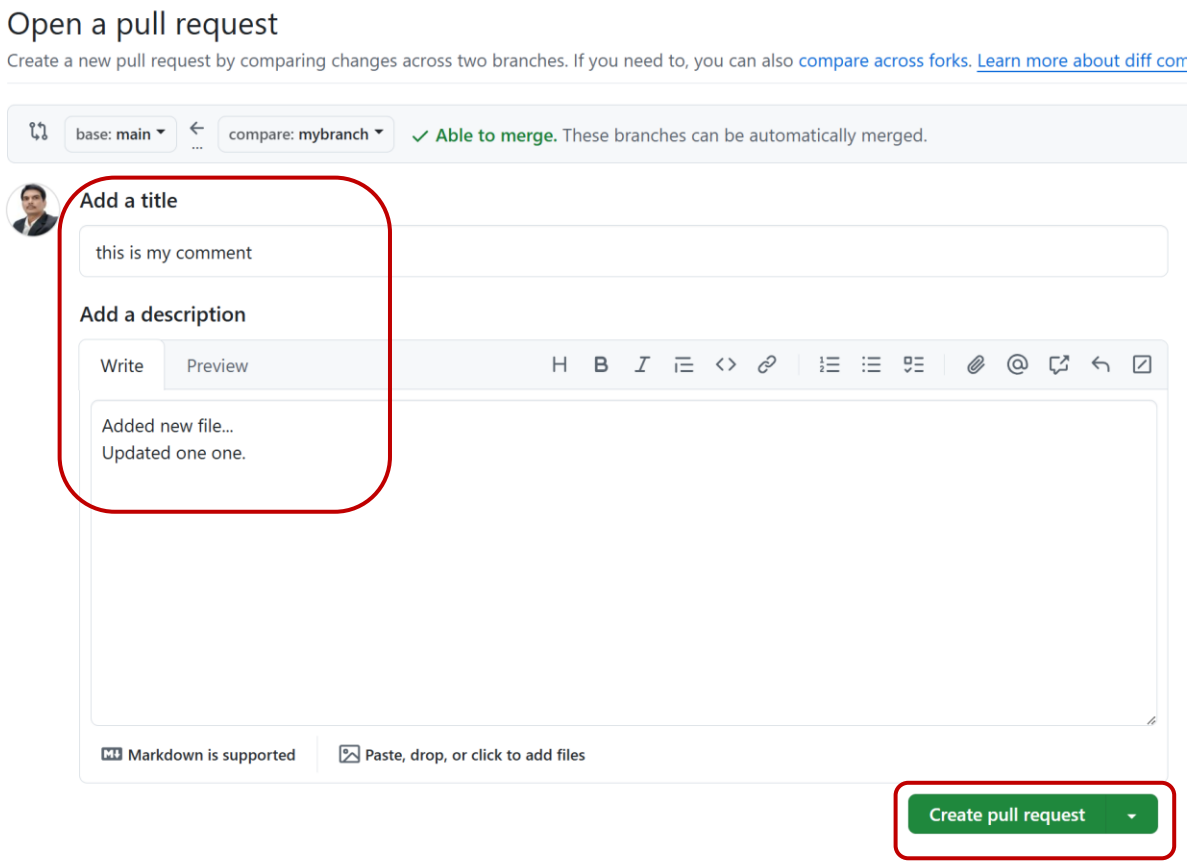


(OR)



3. Click **Compare & pull request / New pull request**.
4. Enter a meaningful title and description.
5. Click **Create pull request**.





Step 7: Code Review and Feedback

- Collaborators will review the pull request and may leave comments or request changes.
- If changes are requested:
    - Make the necessary edits in your local branch.
    - Stage and commit the changes:

      ```
      git add .
      ```

      ```
      git commit -m "Updated file1.txt based on review feedback"
      ```

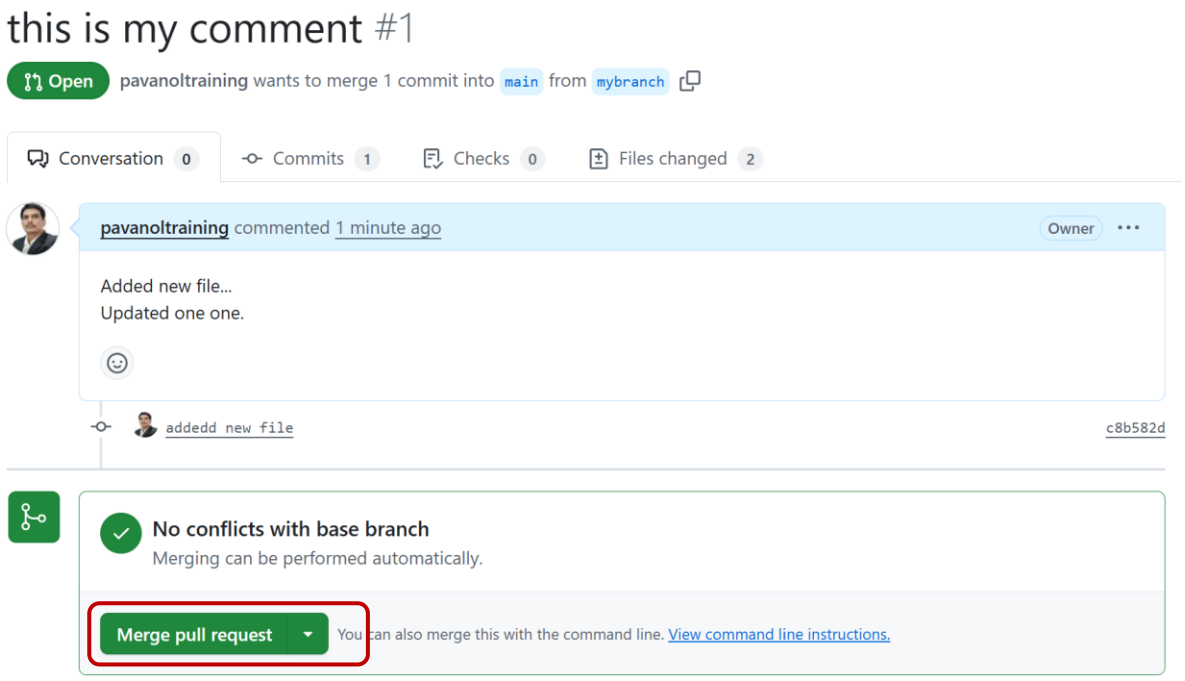
    - Push the changes again:

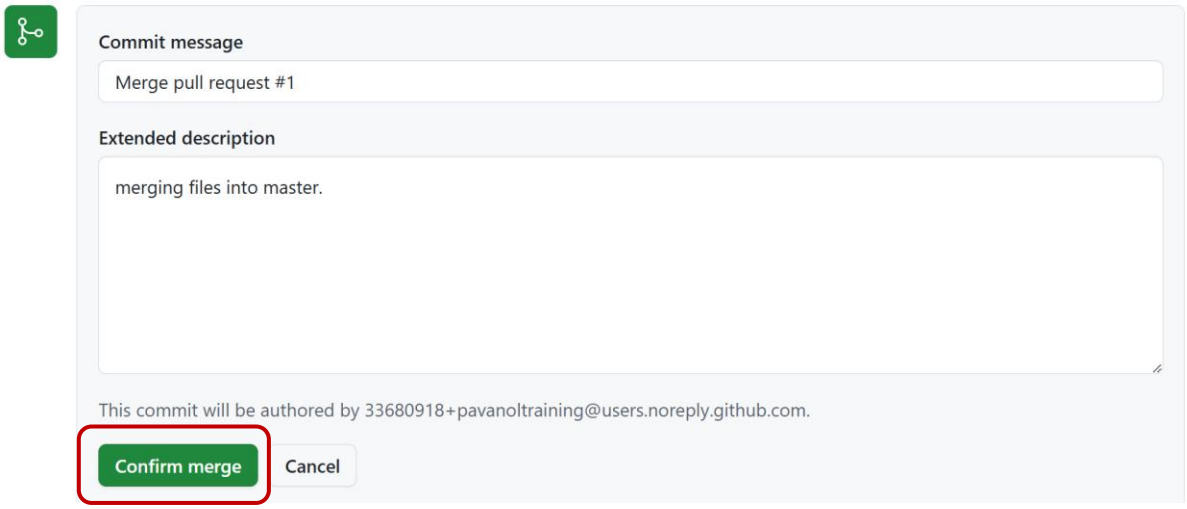
      ```
      git push origin mybranch
      ```

Step 8: Merge the Pull Request to master

Once the pull request is approved:

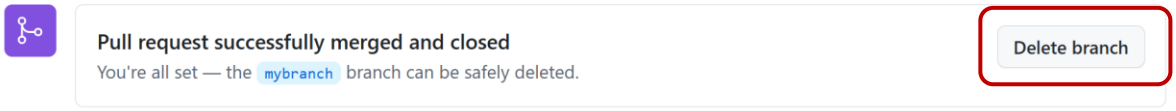
1. In GitHub, click **Merge pull request**.
2. Confirm the merge.
3. Your branch will be merged into master.





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 mybranch
git push origin --delete mybranch
```

Git Tutorials:

https://tinyurl.com/2ktjbscr