

# Guide to Git Commands, Open-Source Contributions, Creating Pull Requests

By Garvit Singh, IT Sophomore

## Git Commands

1. `git init` - To initialise an empty git repository in a directory.
2. `git status` - To see the status of tracked and untracked files and whether they are connected or not.
3. `git add` - To add files to the main branch in order to commit them.
4. `git commit` - To commit the files. Use the `-m` switch to provide a commit message.
5. `git log` - To see history of all commits you've made.
6. `git reset` - Deletes all commits above the specified Hash ID.

7. git stash - When you dont want to commit the changes but rather save them for future. It basically sends the files to the backstage to be worked upon.
8. git stash pop - To bring the stashed files back from backstage.
9. git stash clear - To permanently delete files which are stashed.
10. git remote add origin - To bring a repository into your terminal, origin being the name of the repository.
11. git remote -v : Shows all URLs associated with the repository.
12. git push origin main - To push changes into the main branch of the origin repository.
13. git remote set-url origin - Change URL of the origin remote repository.
14. git clone - Clone a forked repository
15. git remote add upstream - Add URL of the original repository which has been forked.
16. git branch - Create a new branch

17. git checkout - Bring the head to the specified branch.
18. One branch can open only one pull request.
19. git config --global user.email - Configures the global email associated with your Git identity.
20. git config --global user.name - Configures the global username associated with your Git identity.
21. Bringing a forked project even with the main project.

1. First Method

- git checkout main : This switches your local repository to the main branch.
- git fetch --all --prune : This fetches all branches and tags from the remote repository and prunes deleted references.
- git reset --hard upstream/main : This sets your main branch to the same state as the upstream/main branch.

upstream is typically a remote that points to the main repository you forked from.

- `git push origin main`(main branch of local system) : This pushes the changes to your fork's main branch on the remote repository.

## 2. Second Method

- `git pull upstream main` : This command does two things. First, it fetches changes from the upstream/main branch and merges them into your current local main branch. It's equivalent to running `git fetch upstream main` followed by `git merge upstream/main`. This assumes you've already set up the upstream remote.
- `git push origin main` : This pushes the merged changes to your fork's main branch on the remote repository.

22. `git rebase -i` : To merge different commits into one. Here, you have to assign either 'pick' or 's' to the commit. The commits having 's'

will be merged into the first upper commit that has 'pick'.

## 23. Sudo Commands for Debian family of Linux Distros

- `sudo apt-get update` - Updates your cache
- `sudo apt-get upgrade` - Upgrades all the software on the system
- `sudo apt-get remove {package_name}`
- `sudo apt-get purge {package_name}` - Remove a file
- `sudo apt-get autoremove` - Removes unwanted dependencies
- `sudo apt-get clean` - Deletes all compressed package archives.

# Creating Pull Requests on GitHub - Step by Step Process

Creating a pull request (PR) on Git involves several steps. Here's a step-by-step procedure to create a pull request:

## 1. Fork the Repository

- Go to the GitHub repository you want to contribute to.
- Click the "Fork" button in the upper-right corner. This creates a copy (fork) of the repository in your GitHub account.

## 2. Clone Your Fork

- Open a terminal or Git client on your local machine.
- Clone your forked repository to your local machine using the following command:
  - `git clone {fork-url}`

## 3. Create a New Branch

- Create a new branch for your changes. This is a good practice to isolate your work from the main branch. Use the following

commands:

- `git checkout -b feature-branch-name`

#### **4. Make Changes**

- Make the necessary changes to the code in your local branch.

#### **5. Commit Changes**

- Add your changes to the staging area:
  - `git add .`
- Commit your changes with a meaningful message:
  - `git commit -m "Your commit message here"`

#### **6. Push Your Branch to GitHub**

- Push your branch to your fork on GitHub:
  - `git push origin {feature-branch-name}`

#### **7. Create the Pull Request**

- Go to your fork on GitHub and select the branch you just pushed.

- Click the "New Pull Request" button.
- In the "base" or "target" repository and branch, select the repository and branch you want to merge your changes into. Typically, this will be the original repository's main branch.
- In the "compare" or "head" repository and branch, select your fork and branch with the changes.
- Provide a descriptive title and description for your pull request, explaining what it does and why.
- If your pull request is related to an issue in the repository, you can reference it using the # symbol and the issue number (Ex - "Closes #123").
- Review your changes and, if everything looks good, click the "Create Pull Request" button.

## **8. Discuss and Review**

- Your pull request will now be open for discussion and review by the project maintainers and collaborators.



- Be prepared to address feedback and make additional changes as needed.

## **9. Merge the Pull Request**

- Once your pull request is approved, a maintainer or collaborator can merge it into the main repository.

## **10. Clean Up**

- After your pull request is merged, you can delete the branch both locally and on your fork on GitHub.

Your pull request is now part of the main project, and your contributions have been successfully integrated. This process can vary slightly depending on the Git platform and the specific repository, but the general steps are consistent.

# Contributing To Open-Source - Step By Step Process

Contributing to open-source projects is a great way to gain experience and give back to the community. Here's a step-by-step procedure to get started with contributing to open-source projects

## 1. Select a Project

- Choose an open-source project that interests you and aligns with your skills and expertise. You can explore platforms like GitHub, GitLab, and Bitbucket to find projects.

## 2. Set Up Version Control

- If you haven't already, install Git on your local machine.
- Configure Git with your name and email address using:
  - `git config --global user.name "git_username"`
  - `git config --global user.email "git_email"`

## 3. Create an Account on the Hosting Platform

- Most open-source projects are hosted on platforms like GitHub. Create an account on the platform where the project is hosted.

#### **4. Fork the Repository**

- Go to the project's repository on the hosting platform (e.g. GitHub).
- Click the "Fork" button to create your own copy of the project in your GitHub account.

#### **5. Clone Your Fork**

- Clone your forked repository to your local machine using the following command:
  - `git clone {fork-url}`

#### **6. Set Up Upstream Remote**

- Add the original project repository as an "upstream" remote to keep your fork up to date. Use the following command:
  - `git remote add upstream {original-repo-url}`

## 7. Create a Feature Branch

- Create a new feature branch for your contribution. It's a good practice to isolate your work. Use the following commands:
  - `git checkout -b feature-branch-name`

## 8. Make Changes

- Make the necessary changes to the code, following the project's contribution guidelines and coding standards.

## 9. Commit Changes

- Add your changes to the staging area:
  - `git add .`
- Commit your changes with a descriptive message:
  - `git commit -m "Your commit message here"`

## 10. Push Your Branch to Your Fork

- Push your branch to your fork on the hosting platform:
  - `git push origin {feature-branch-name}`

## **11. Create a Pull Request**

- Go to your fork on the hosting platform.
- Click the "New Pull Request" button.
- Select the target branch in the original project repository where you want to merge your changes.
- Provide a title and a detailed description of your changes.
- If your pull request addresses a specific issue, reference it by using # (Ex - "Closes #123").
- Submit the pull request.

## **12. Discuss and Review**

- Be prepared to engage in discussions with project maintainers and other contributors.
- Address feedback and make additional changes as needed.

## **13. Merge Your Contribution**

- Once your pull request is approved, a project maintainer will merge your changes into the original repository.

#### **14. Keep Your Fork Updated**

- Periodically sync your fork with the original repository to ensure your local and remote branches are up to date.

#### **15. Continue Contributing**

- Congratulations! Your contribution has been successfully merged. Continue contributing to the project and explore other open-source projects that interest you.

Remember that each project may have its specific contribution guidelines and processes, so always check the project's documentation or CONTRIBUTING.md file for any project-specific instructions.

Thanks For Reading! ❤️



**By GARVIT SINGH**

Information Technology