

Here's Everything About Git Push and Pull

[ADVANCED](#)[GITHUB](#)

Introduction

Git, a distributed version control system, plays a crucial role in modern software development by enabling efficient collaboration among developers. It helps manage and track changes in your code, which is as easy as a breeze. At the heart of this version control system lies the dynamic duo of “git pull” and “git push,” two commands that serve as the linchpin connecting developers across the digital landscape.

Join me on a journey through the intricate tapestry of Git’s pull and push commands as we unravel the secrets behind their collaborative magic and explore how they foster a harmonious environment for code collaboration.

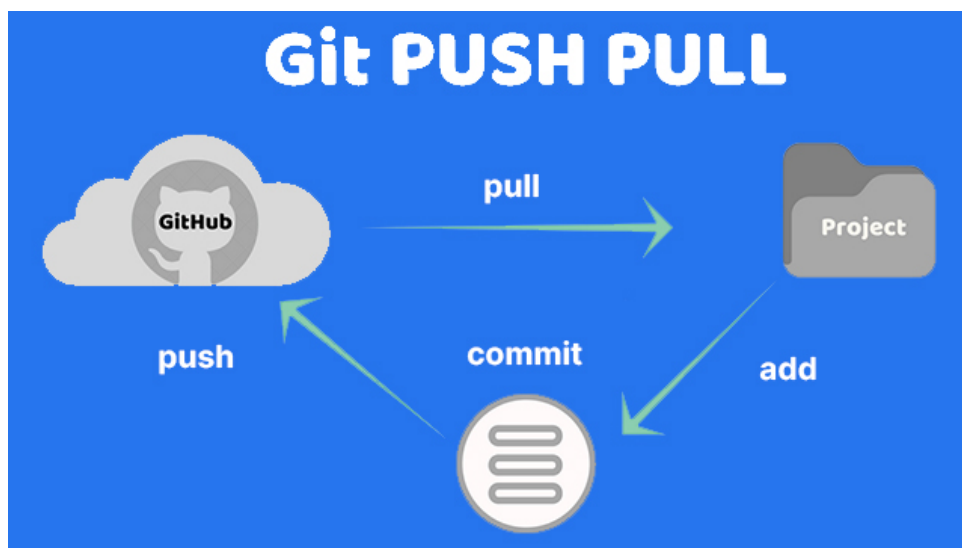


Table of contents

- [Understanding Git Push and Pull](#)
 - [Git Push](#)
 - [Git Pull](#)
- [Deep dive into Git Push](#)
 - [How to use Git Push using Command Line?](#)
 - [How to Utilize GitHub Desktop for Git Push?](#)
- [Deep dive into Git Pull](#)
 - [git fetch](#)
 - [git merge](#)
 - [How to use Git Push Using Command Line](#)

- [How to use Git Push Using Github Desktop:](#)
- [Best Practices for Using Git Push and Pull](#)

Understanding Git Push and Pull

Git Push

Git Push is a command that uploads local repository changes to the remote repository. This ensures that the latest code changes are accessible to team members, promoting a unified and up-to-date codebase.

It's like shouting, "Hey, world! Check out my awesome code!"

Git Pull

Git Pull is a command used to fetch and integrate changes from a remote repository into the local repository. It helps to keep the local copy synchronized with the remote, preventing divergence in the codebase.

Deep dive into Git Push

Let's deep dive into the Git Push command, covering understanding the command, its use, common scenarios, and examples, and troubleshooting common issues.

Syntax

The basic syntax of `git push` is:

```
git push <remote-name> <branch-name>
```

remote-name

The name of the remote repository you want to push to (e.g., origin).

Branch-name

The name of the branch you want to push.

How to use Git Push using Command Line?

Create a New Repository

To initiate a new repository, click on the plus sign and provide essential details such as the repository name and description.

Launch Git Bash

Obtain Git Bash from [[Downloads](#)]. It serves as a shell interface, aligning with UNIX commands to interact with the operating system.

Initialize the Git Repository

Execute 'git init' to initialize a new repository. This command establishes a new empty repository with a hidden '.git' directory at the project's top level, consolidating all revision information.

Add Files to the Local Repository

- Add all files to the designated folder using 'git add .' in your Git Bash.
- Verify staged file status for the initial commit by employing 'git status.'

Commit Staged Files with a Message

- Craft a commit message through 'git commit -m 'your message"' to integrate changes into the local repository.
- The 'git commit' command with the '-m' flag allows concise, imperative sentence messages (up to 50 characters) detailing the nature and reason for the change.

Copy Remote Repository URL from GitHub

Copy the HTTPS or URL from your GitHub account, representing the location of the remote repository.

Add Copied URL as Remote Repository

Execute 'git remote add origin 'your_url_name"' to link the remote repository. Here, 'origin' is the remote name, and the provided URL is from GitHub.

Push Local Repository Code to GitHub

Use 'git push -u origin master' to push local content to GitHub. The 'origin' is the default remote repository name, '-u' denotes upstream (similar to '-set-upstream'), and 'master' is the branch name.

View Files in Your GitHub Repository

Explore your hosted files on GitHub to confirm the successful upload.

How to Utilize GitHub Desktop for Git Push?

GitHub Desktop is a versatile tool compatible with various operating systems, providing a user-friendly Graphical User Interface (GUI) for seamlessly pushing local content from a local repository to a remote repository, such as GitHub.

To initiate the process, first, open your GitHub account in a web browser.

Select “Set up in a Desktop”

Click on the designated button, triggering a pop-up window. Proceed by selecting “Open GitHub Desktop.”

Cloning in GitHub Desktop

Utilize the “Clone” button, as illustrated below. Upon cloning, a new folder is generated on your local computer, housing a concealed “.git” directory.

Transfer Essential Files to the Clone Folder

Copy all necessary files, including images and README files, from your local computer into the designated clone folder.

Commit to Master in GitHub Desktop

Files added to the clone folder are visible in GitHub Desktop. Conclude by crafting a message and executing the “Commit to master” action.

Publish Branch in GitHub Desktop

Click on “Publish Branch” to upload all local content to GitHub, ensuring synchronization between repositories.

Upon completing these steps, you can navigate to your GitHub repository to review the uploaded content.

Deep dive into Git Pull

Git Pull is a Git command used to fetch and integrate changes from a remote repository into your current working branch. It essentially combines two actions: `git fetch` and `git merge`.

`git fetch`

Retrieves changes from the remote repository but doesn't merge them into your working branch.

git merge

Integrates the fetched changes into your working branch.

Syntax

```
git pull [options] [remote] [branch]
```

Options

Optional flags that modify the behavior of the `git pull` command.

Remote

The remote repository from which you want to pull changes. If you have cloned a repository, the remote is usually named "origin" by default.

Branch

The branch from which you want to pull changes. If you don't specify a branch, Git will use the currently checked-out branch.

How to use Git Push Using Command Line

Fork the Repository

A "Fork" refers to creating a duplicate of a repository. This duplication allows you to experiment freely with modifications without impacting the original project. (Source)

Open your bash on your computer

Navigate to the desired path or folder using the `cd` command. Utilize the `ls` command to view the directory contents, revealing the presence of the 'README.md' file in this case.

Create a new branch

Generate a new branch using the command `git checkout -b 'branch_name'`. In this code, the '-b' flag initiates a new branch, while 'branch_name' assigns a specific name to the branch. The `checkout` command facilitates the switch to the newly created branch.

Push the repository to GitHub

Push the content using ``git push origin 'branch_name'``.

In this code, 'origin' represents the remote repository, and 'branch_name' is the designated branch for uploading local content.

Pull request for a specific branch on GitHub

- Visit your [GitHub](#) repository to confirm the presence of the new branch.
- Proceed to step 8. Before that, update your local repository with the upstream repository. Refer to a detailed blog on “How To Create a Pull Request on GitHub” for guidance.
- Alternatively, use the command line with ``git pull-request`` to complete the Pull Request, forcibly pushing your current branch to the remote repository.

Open a Pull request

- Click the “Create pull request” button to finalize the action.

Deleting a Branch after the Pull Request is Merged

- Navigate to the main repository page and select “Pull requests.”
- Click ‘Closed’ to view a list of all made Pull Requests. Select the one related to your branch for deletion.
- Click ‘Delete branch’ to execute the action.

How to use Git Push Using Github Desktop:

Follow the steps below to create a PULL request using GitHub Desktop.

Clone and Open in Desktop

Clone the project and select “Open in Desktop.”

Create a New Branch

Generate a new branch named “fix-typo-imp.”

Edit the imp File in the Text Editor

Modify the imp file by correcting the typo and adding necessary text.

Commit the Changes

Compose a commit message and click “Commit to fix-typo-imp.”

Publish the Branch

Publish the branch to push the commit to GitHub.

Initiate a PULL Request

Click “Create pull request” to begin the PULL request process. Optionally, add a message and click “Create pull request” again.

Follow the subsequent steps as outlined in the “PULL Request through Command Line.”

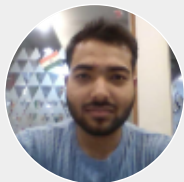
Best Practices for Using Git Push and Pull

1. **Pull First:** Always pull before you push to avoid conflicts and ensure you have the latest changes.
2. **Branch Cleanliness:** Use feature branches for your work, keeping the main branch clean and stable.
3. **Commit Frequently:** Make small, focused commits with clear messages to track changes effectively.
4. **Rebase for Clarity:** Use rebase instead of merge for a cleaner commit history, reducing unnecessary merge commits.
5. **Test Locally:** Before pushing, test your changes locally to catch issues early and ensure a smooth integration.
6. **Review Changes:** Review your code changes before pushing to maintain code quality and catch errors.
7. **Descriptive Commits:** Write meaningful commit messages to communicate changes effectively.
8. **Use SSH Keys:** Set up SSH keys for secure and convenient authentication when pushing and pulling.
9. **Gitignore Files:** Use a .gitignore file to exclude unnecessary files and prevent them from being pushed.
10. **Communication:** Communicate with your team about your changes and coordinate before pushing major updates.

Conclusion

Mastering Git push and pull commands is crucial for effective version control and collaboration in any project. These commands, along with other Git operations, provide a robust framework for managing and tracking changes. So, the next time you’re working on a project, remember to pull before you push, commit your changes regularly, and keep exploring the vast capabilities of Git. Happy coding!

Ready to revolutionize your AI & ML journey? Enroll in our [Certified AI & ML BlackBelt Plus Program](#) and supercharge your career! Experience 1:1 mentorship, hands-on guided projects, and a personalized learning path tailored to your goals. Crush doubts with on-demand clearing sessions. Level up in AI & ML – power ahead with confidence!



[Aayush Tyagi](#)