1. Difference Between Git and GitHub

**Git**: Git is a distributed version control system for tracking changes in source code during software development. It is designed for coordinating work among programmers, but it can be used to track changes in any set of files. Its goals include speed, data integrity, and support for distributed, non-linear workflows.

GitHub: It is a web-based Git repository. This hosting service has cloud-based storage. GitHub offers all distributed version control and source code management functionality of Git while adding its own features. It makes it easier to collaborate using Git. Additionally, GitHub repositories are open to the public. Developers worldwide can interact and contribute to one another’s code, modify or improve it, making GitHub a networking site for web professionals.

The main Git vs GitHub difference is in their functionality. While they both provide source code management and make merging and sharing code easier, this is pretty much where their similarities end. Think of Git as a single computer and GitHub as a network of multiple interconnected computers.

Git is a free, open-source software distributed version control system (DVCS) designed to manage all source code history. It can keep a history of commits, can reverse changes, and lets developers share code. Each developer must have git installed on his or her local device to collaborate. It is commonly referred to as one of the best development tool to understand and use in the developer space, and it’s among the most widely used tools today.

GitHub, on the other hand, is a web-based hosting service for Git repositories. It offers all of Git’s DVCS SCM and has some additional features. This includes collaboration functionality like project management, support ticket management, and bug tracking. With GitHub, developers can share their repositories, access other developers’ repositories, and store remote copies of repositories to serve as backups.

1. Basic Git Commands

* git init: To create a new repo, you'll use the git init command. git init is a one-time command you use during the initial setup of a new repo. Executing this command will create a new .git subdirectory in your current working directory.
* git config: The git config command lets you configure your Git installation from the command line. This command can define everything from user info, to preferences, to the behavior of a repository.

Git stores configuration options in three separate files, which lets you scope options to individual repositories (local), user (Global), or the entire system (system).Define the author name to be used for all commits in the current repository.

git config --global user.name Saloni

* git clone: The git clone command is used to create a copy of a specific repository or branch within a repository.  This is only done once, when you begin working on a project, and would follow the syntax of git clone [URL].
* git add: The git add is a command, which adds changes in the working directory to the staging area. With the help of this command, you tell Git that you want to add updates to a certain file in the next commit. But in order to record changes, you need to run [git commit](https://www.w3docs.com/learn-git/git-commit.html) too.
* git commit: git commit creates a commit, which is like a snapshot of your repository. These commits are snapshots of your entire repository at specific times. You should make new commits often, based around logical units of change. Over time, commits should tell a story of the history of your repository and how it came to be the way that it currently is. Commits include lots of metadata in addition to the contents and message, like the author, timestamp, and more.

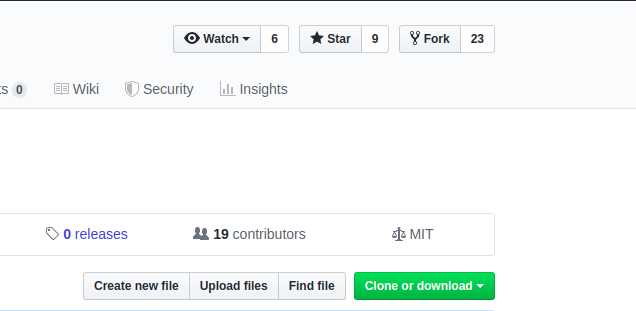
1. How to make a Pull Request?

Pull requests are the way we contribute to group projects or open source projects.

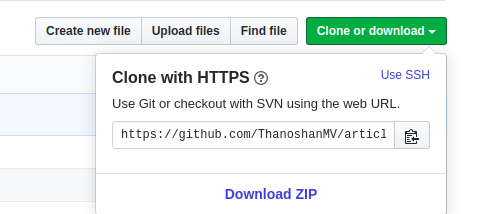
For instance, a user Harry forks a repository of ThanoshaMV and makes changes to that repository. Now Harry can make a pull request to ThanoshaMV, but it’s up to ThanoshaMV to accept or decline it. It’s like saying, “ThanoshaMV, would you please pull my changes?”

### 1. Fork the repository

Fork the repository by clicking the fork button on the top of the page. This will create an instance of that entire repository in your account.

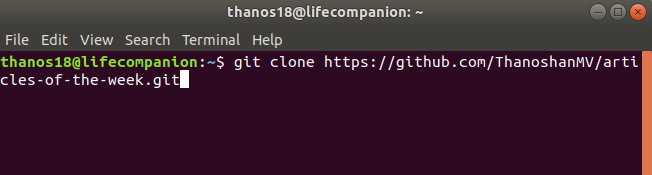


### 2. Clone the repository

Once the repository is in your account, clone it to your machine to work with it locally. To clone, click on the clone button and copy the link.

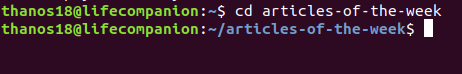
Open the terminal and run the following command. It will clone the repository locally.

$ git clone [HTTPS ADDRESS]



Now we have set up a copy of the master branch from the main online project repository.We need to go to that cloned directory by running this command:

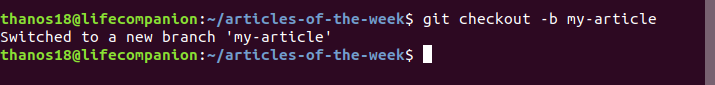
$ cd [NAME OF REPOSITORY]



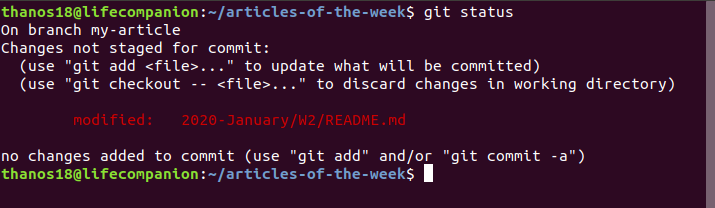
### 3. Create a branch

It’s good practice to create a new branch when working with repositories, whether it’s a small project or contributing to a group's work. Branch name should be short and it should reflect the work we’re doing. Now create a branch using the git checkout command:

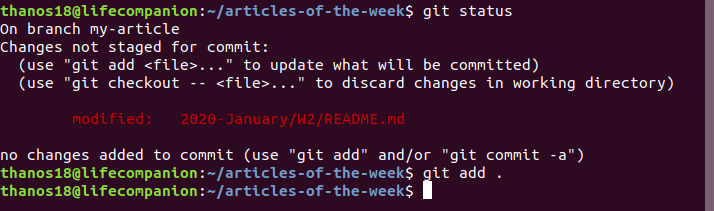
$ git checkout -b [Branch Name]



4. Make changes and commit them. Make essential changes to the project and save it. Then execute git status , and you’ll see the changes.

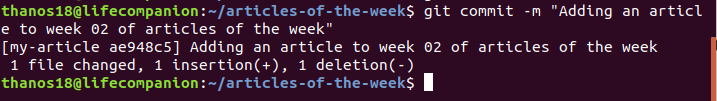


Add those changes to the branch you just created using the git add command:

$ git add . 

Now commit those changes using the git commit command:

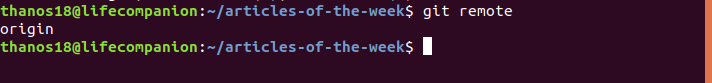
$ git commit -m "Adding an article to week 02 of articles of the week"



### 5. Push changes to GitHub

In order to push the changes to GitHub, we need to identify the remote’s name.

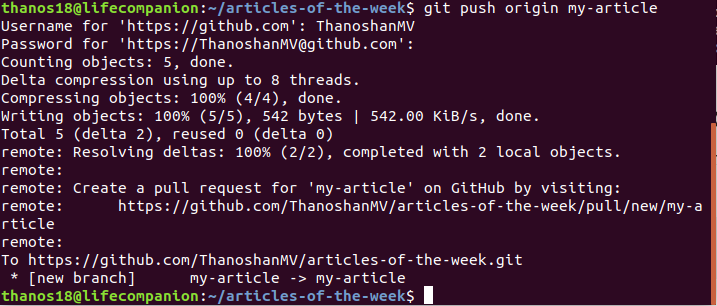
$ git remote



For this repository the remote’s name is “origin”.

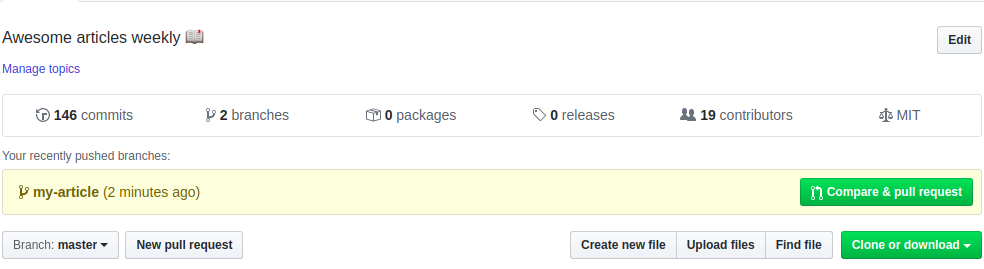
After identifying the remote’s name we can safely push those changes to GitHub.

git push origin [Branch Name]



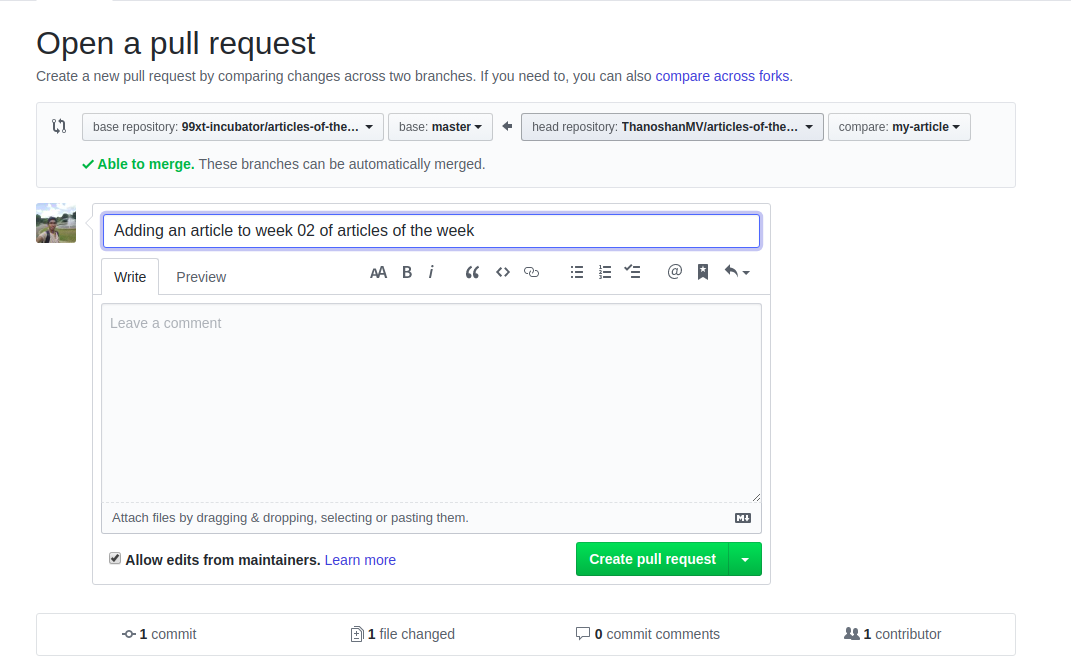
### 6. Create pull request

Go to your repository on GitHub and you’ll see a button “Compare & pull request” and click it.



Please provide necessary details on what you’ve done (You can reference issues using “#”). Now submit the pull request.

Congratulations! You've made your first pull request.



If your pull request is accepted you’ll receive an email.

### 7. Sync your forked master branch

Before submitting any pull requests to the original repository you have to sync your repository to the original one.

Even if you are not going to submit a pull request to the original repository, it’s better to sync with the original repository as some additional features and bug fixes may have been done since you forked the original repository.