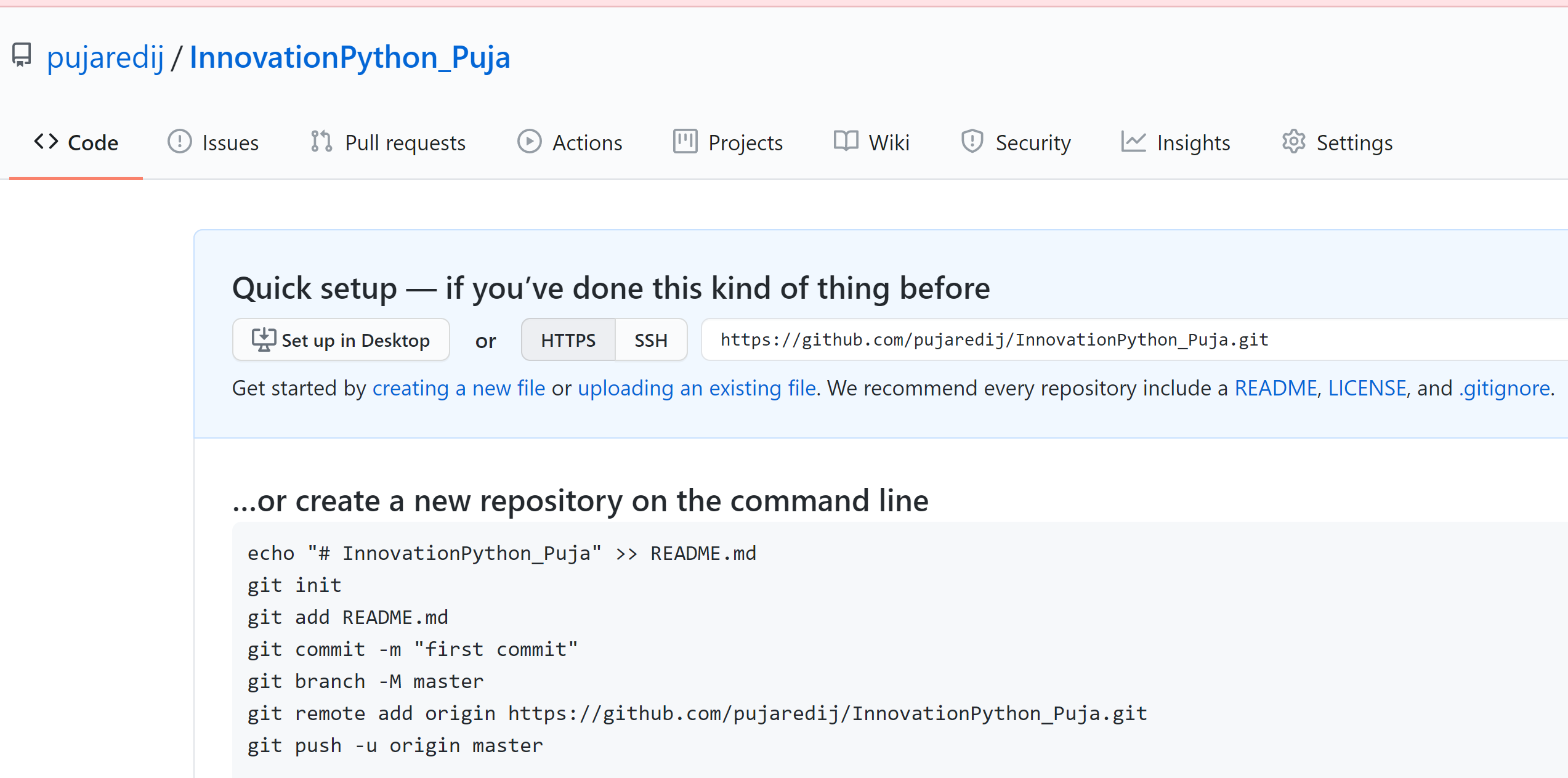
1. Make a repository on GitHub with the name **“InnovationPython\_yourname”**

eg: “InnovationPython\_Ankush”.



**2. Git** is a revision control system, a tool to manage your source code history.

**GitHub** is a hosting service for Git repositories.

So they are not the same thing: **Git** is the **tool**, **GitHub** is the **service for projects that use Git**.

3. A Git Workflow is a recipe or recommendation for how to use Git to accomplish work in a consistent and productive manner. Git workflows encourage users to leverage Git effectively and consistently. Git offers a lot of flexibility in how users manage changes.

4. There are two types of version control systems: Centralized, Distributed

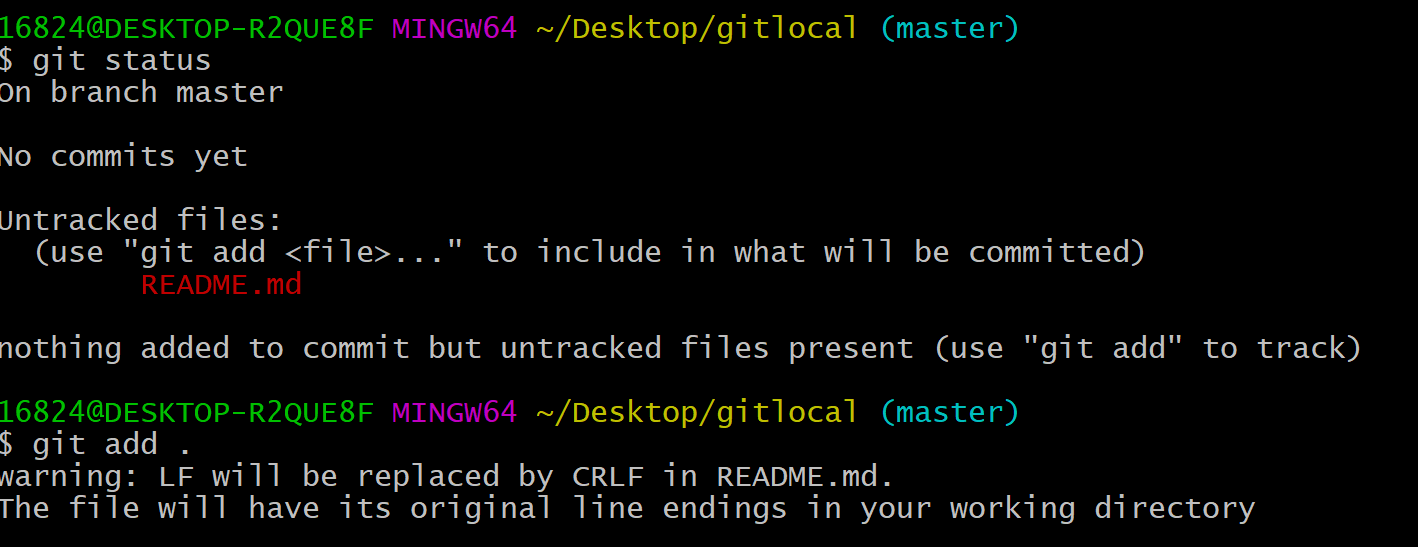
5. Git branches are effectively a pointer to a snapshot of your changes. When you want to add a new feature or fix a bug—no matter how big or how small—you spawn a new branch to encapsulate your changes. This makes it harder for unstable code to get merged into the main code base, and it gives you the chance to clean up your future's history before merging it into the main branch.

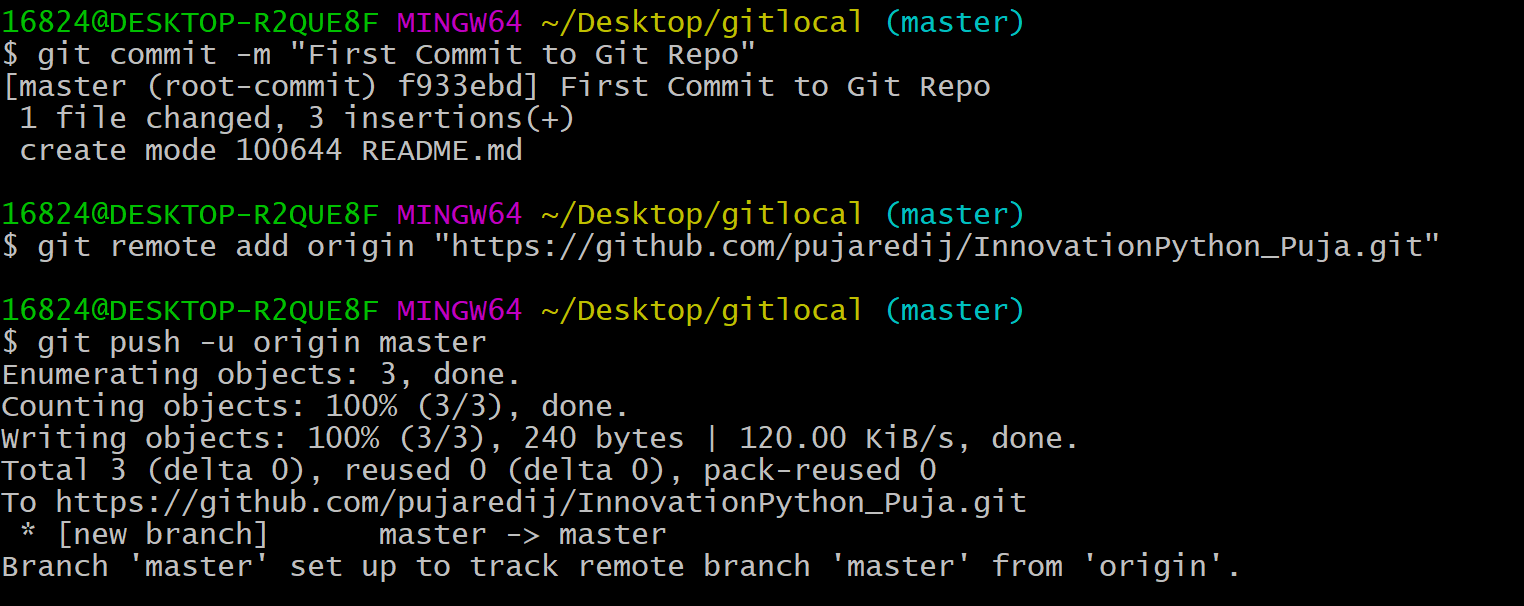
6. Forking Workflow begins with an official public repository stored on a server. But when a new developer wants to start working on the project, they do not directly clone the official repository. Instead, they fork the official repository to create a copy of it on the server.

Initialize an empty Git repository on your local machine with the name **“gitlocal”** and make a **README.md** file in that directory which should contain your name as a heading and a hello message(https://www.makeareadme.com/) .

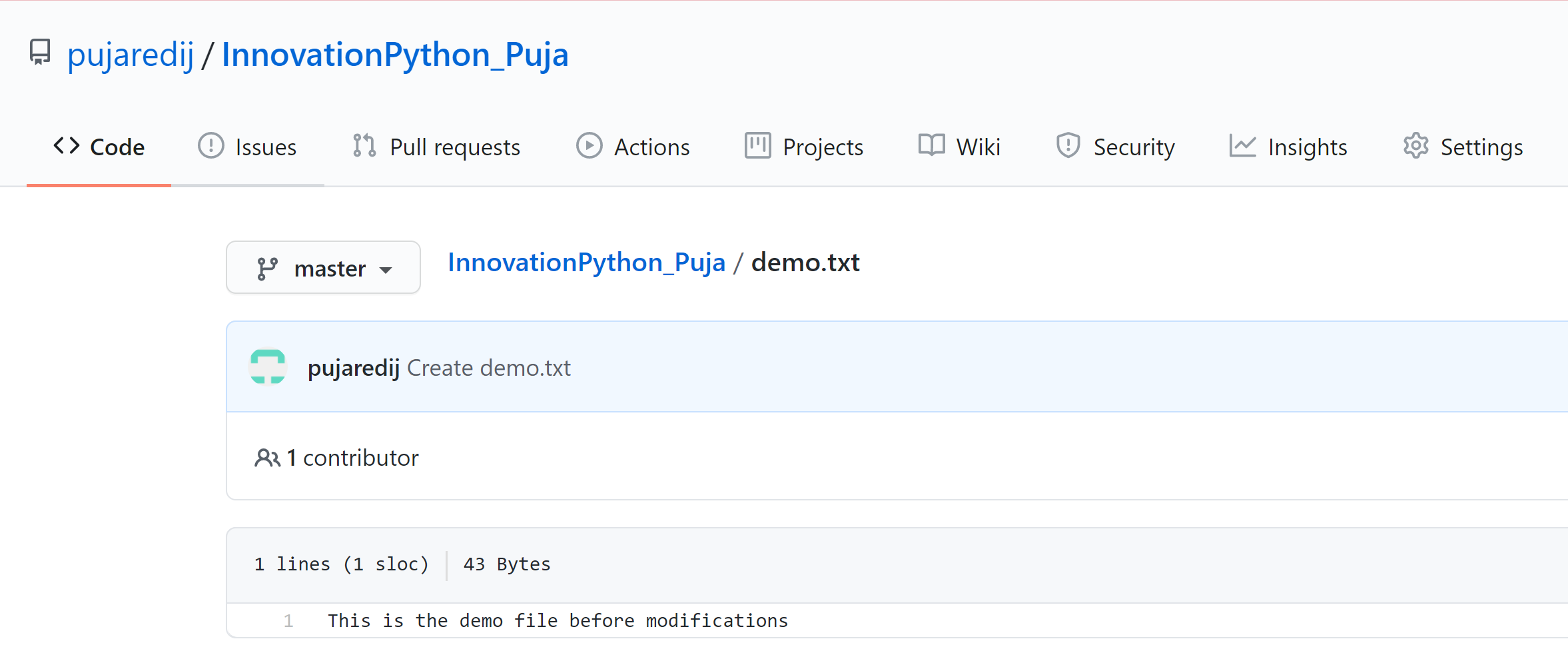


Now check the status of your git directory and push all the files in that directory to your GitHub repo which you have made in the first step. With a massage “First Commit to Git Repo”.

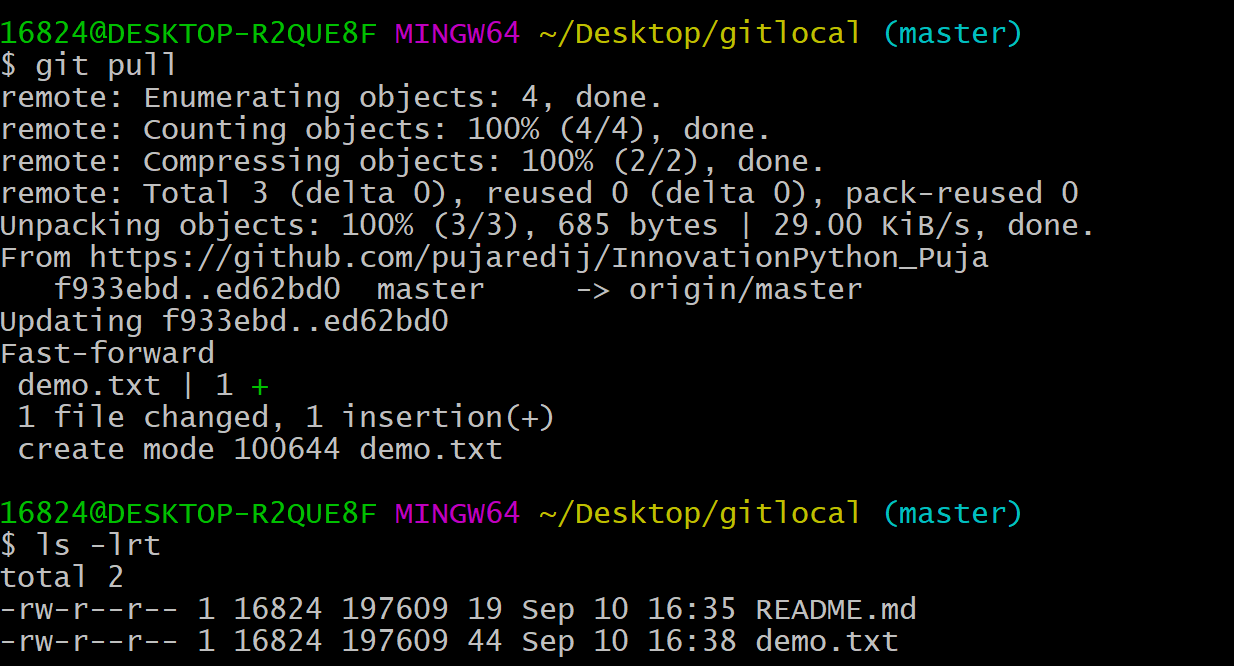




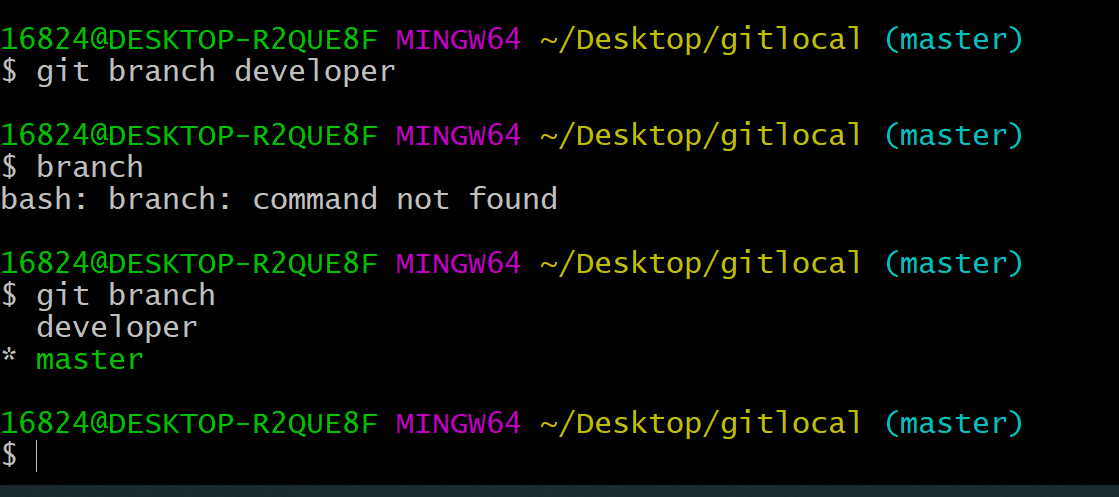
Now add a file to your Github repo named “demo.txt” from the github console with content : “This is the demo file before modifications”.



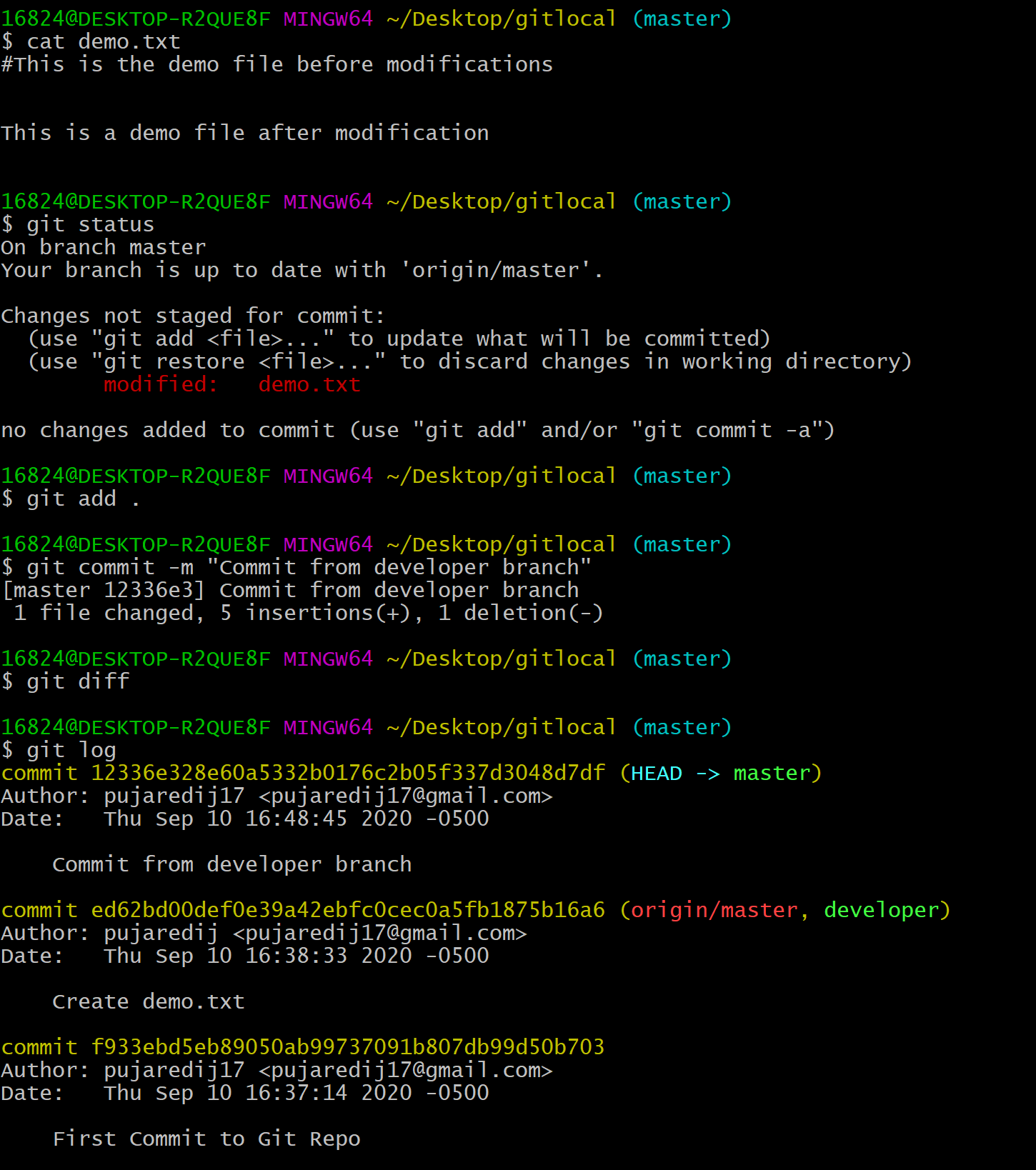
Pull the changes in your git repo to your local machine git directory named **“gitlocal”** and check the status for the modifications done in that repo. This time demo should be visible in your local machine.

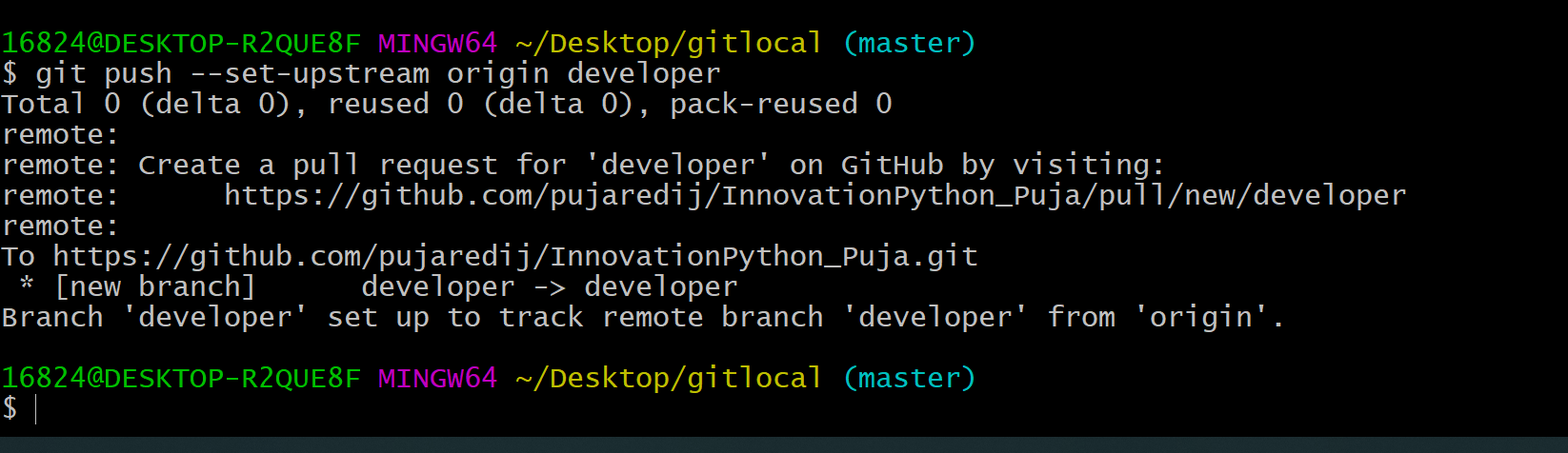


● Now make a new branch in your local machine with the name “**developer**”.

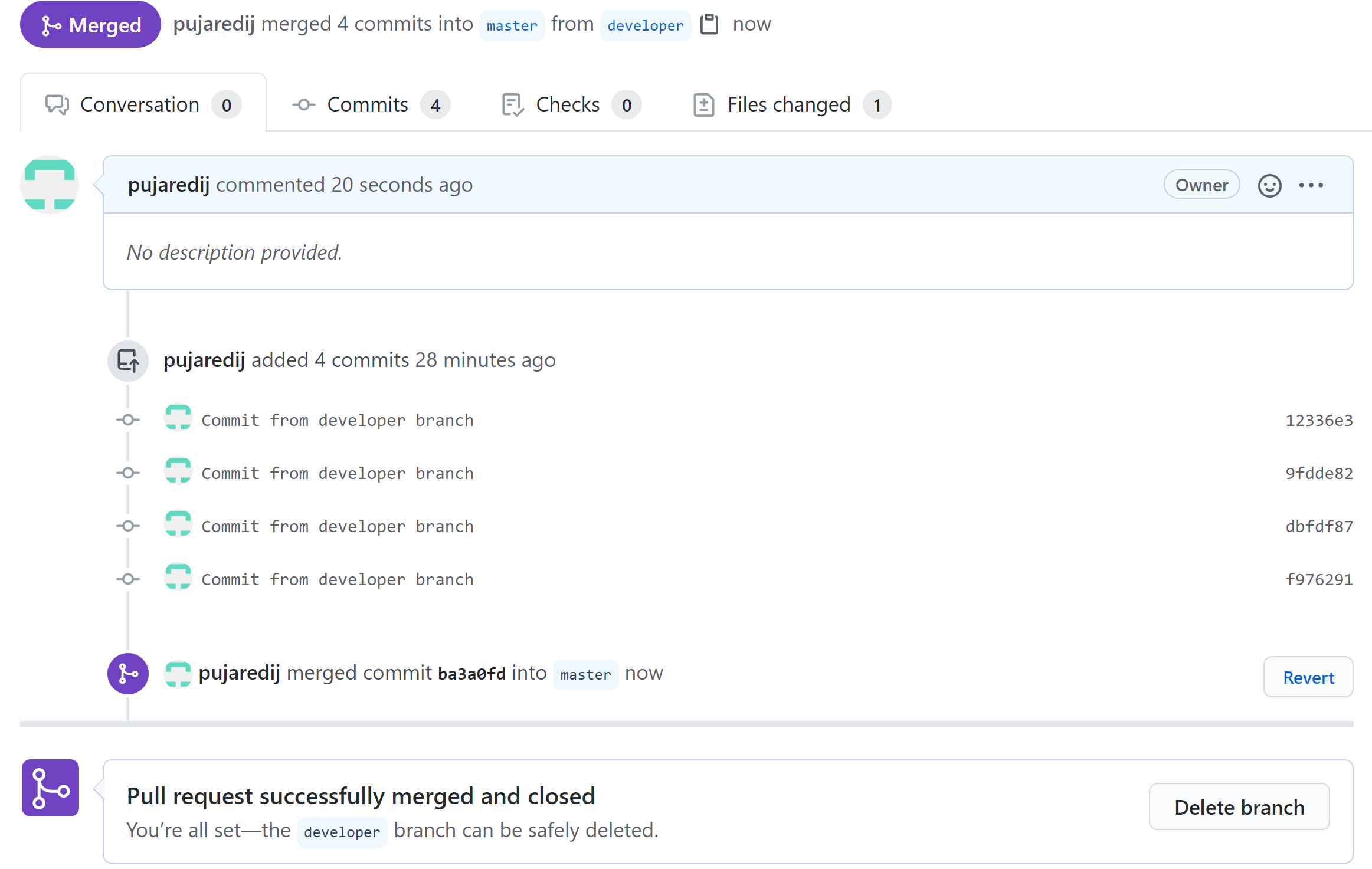


Edit that demo file and write some content in that eg: “This is a demo file after modification” and push the modifications to your GitHub repo from the **developer** branch with a commit message “Commit from developer branch”.





Go to the GitHub console and generate a merge request to master branch after checking the modifications.



At the end delete the developer branch.

