**Navigating GitHub**

1. Sign In
2. Click on Home button

Graphical user interface, application, website

Description automatically generated

1. Switch dashboard content to “FluentCo”

Graphical user interface, text, email, website

Description automatically generated

1. Browse organization’s repositories

Graphical user interface, text, application, email, website

Description automatically generated

1. Go to Teams --> Data Science

A screenshot of a computer

Description automatically generated

1. Go to Repositories

Graphical user interface, application

Description automatically generated

1. You should be able to see the team’s repositories here

Graphical user interface, text, website

Description automatically generated

**Creating New Repository**

1. Click on “+” on top right. “New repository”

Graphical user interface, application, Teams

Description automatically generated

1. Change owner from yourself to FluentCo

Graphical user interface, text, application, email, website

Description automatically generated

1. Choose “Private”
2. Check off “Add a README file”
3. Set .gitignore template: Python
4. Add License: MIT License
5. Click “Create Repository”

Graphical user interface, text, application, email

Description automatically generated

1. Go to Settings in the new repository

Graphical user interface, text, website

Description automatically generated

1. Click “Collaborators and teams”

Graphical user interface, text, application, email

Description automatically generated

1. Click “Add teams” and type “data-science”

A screenshot of a computer

Description automatically generated

1. For now, add as Admin role. Click “Add FluentCo/data-science to this repository”

Graphical user interface, text, application

Description automatically generated

**Creating your token**

1. Go to Settings on the top right of your screen

Graphical user interface, application, Word

Description automatically generated

1. Scroll all the way down to Developer Settings

Graphical user interface, application

Description automatically generated

1. Go to “Personal access tokens”
2. Go to “Tokens (classic)”
3. Click “Generate new token”
4. Click “Generate new token (classic)”

Graphical user interface, text, application, email

Description automatically generated

1. Create a note
2. For now, leave Expiration at 30 days
3. Check everything for scopes
4. Click “Generate token”

Graphical user interface, text, application

Description automatically generatedA screenshot of a computer

Description automatically generated with medium confidence

1. **Once you have your token, save it somewhere safe – you will not be able to retrieve it again**

**Connecting and Testing with Sagemaker**

1. Create a folder called Fluent on Sagemaker

A screenshot of a computer

Description automatically generated with medium confidence

1. Open Launcher
2. Go to Image Terminal. It may take a few minutes to load

A screenshot of a computer

Description automatically generated with medium confidence

Should look like this

Graphical user interface, text

Description automatically generated

1. Go to the Test folder (or whichever folder you need) on GitHub
   1. Navigate to the dedicated folder you have created for the specific project you have created with the command line: cd <Your Folder name>
2. (Go to GitHub) Click “Code”
3. Copy the HTTPS link

Graphical user interface, text, website

Description automatically generated

1. Go to Image Terminal and run the following command:

cd Fluent

**To clone the repository onto sagemaker**

1. You will need to adjust the copied link to include your token that you saved then run it:

git clone https://generated\_token@github.com/FluentCo/Test.git

1. This should create a copy of the GitHub folder into your Fluent folder on sagemaker
   1. To make sure your repository is up to date, run the command: git fetch

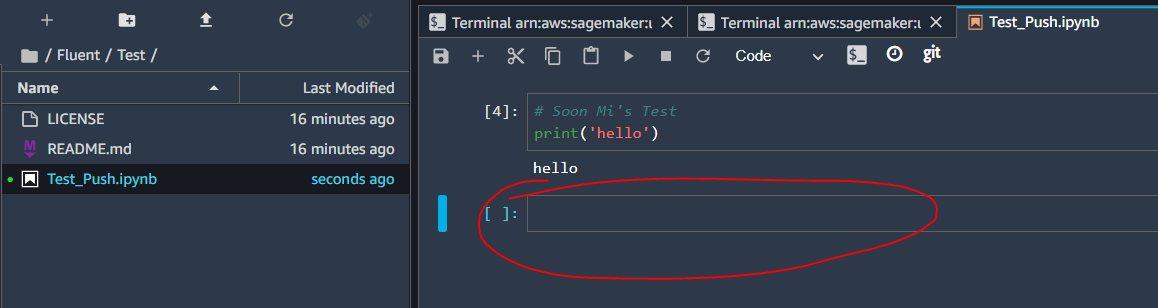
It is usually executed once your repository has been cloned in your sagemaker instance and you revisit the code to make changes but also want to reflect the changes which has been done by you or by someone else in the team.

Graphical user interface, application

Description automatically generated

**To push your changes from sagemaker to GitHub**

1. Create a change in sagemaker. Example, add another line to this notebook
2. Save notebook



1. Go to the folder you want to make changes in (The Repository name which you have cloned)

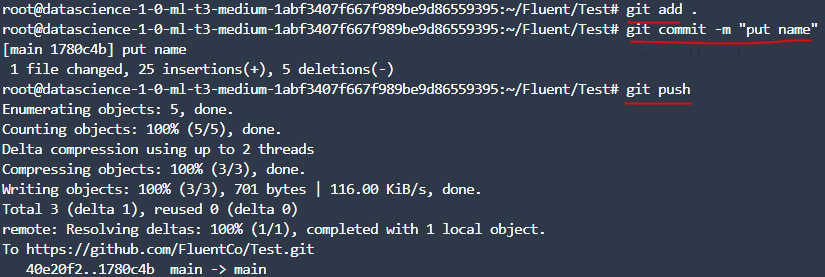
cd Test

1. Use the commands to commit with a note and push back to GitHub. Example, “put name” or “added x’s Test”

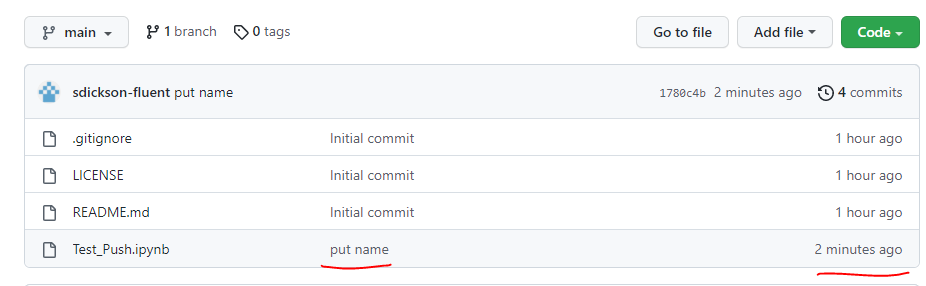
git add .

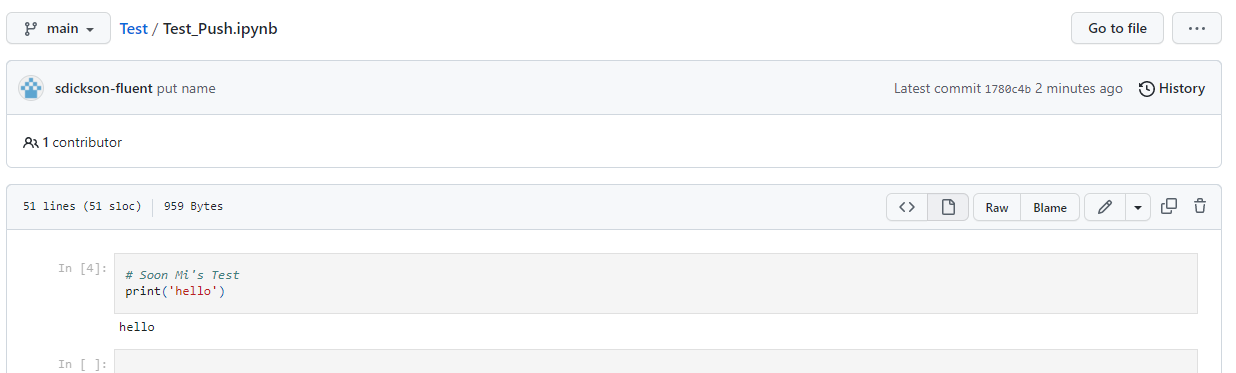
git commit -m “put name”

git push



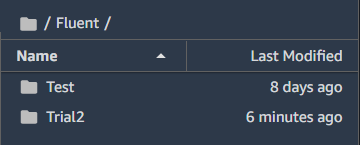
1. When you return to GitHub, the note you stated should be there and the changes should be reflected





**Creating and Merging a New Branch from an Existing Repository**

1. Go to Sagemaker and navigate to the the Git repository of interest in the Image Terminal

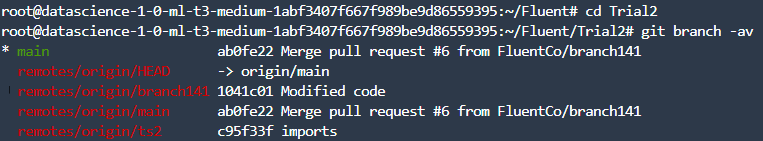


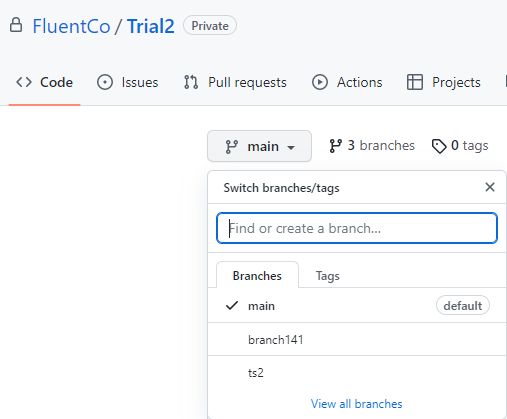
cd Fluent

cd Trial2

1. Check the existing branches. It should match what you see in GitHub

git branch –av





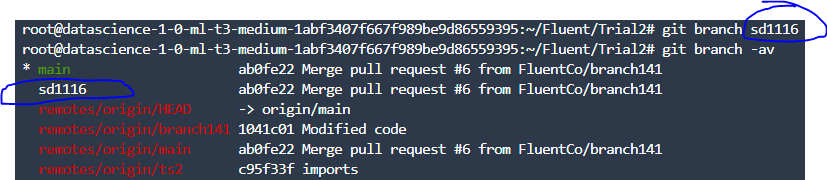
1. Create a new branch. It should show up in GitHub

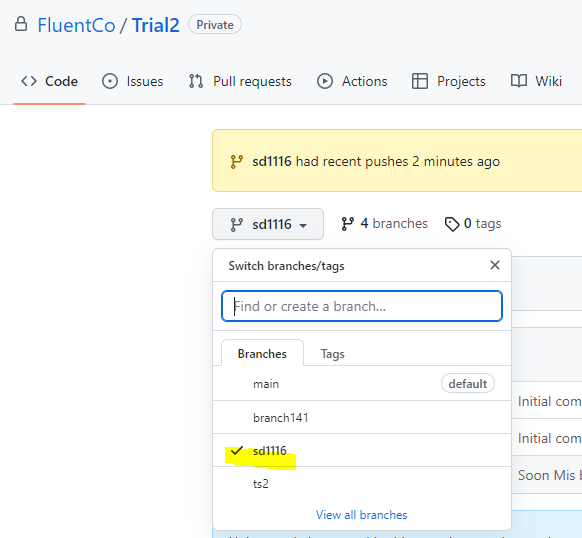
git branch "branch name”



1. Check to make sure your branch is created or check github

git branch –av



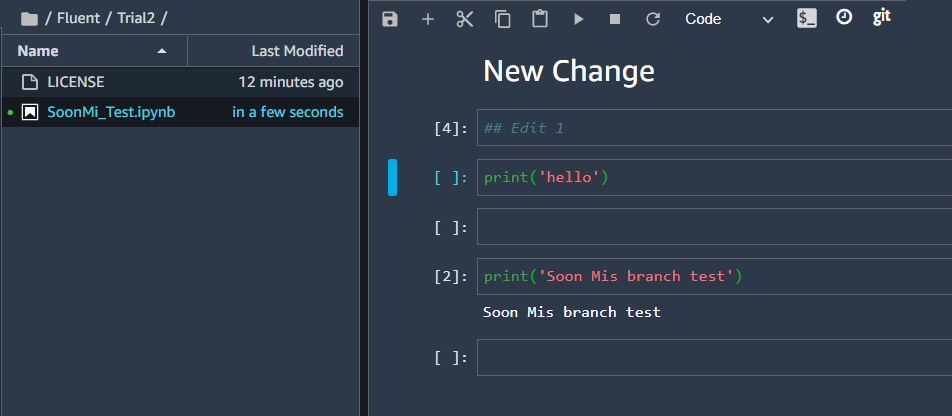


1. Go to the new branch

git checkout “branch name”



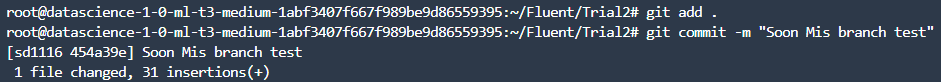
1. Make changes in your code and save



1. Add and commit your changes

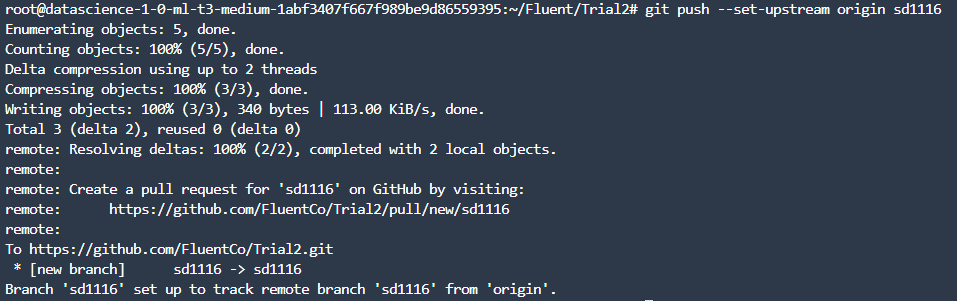
git add .

git commit –m “add comment”

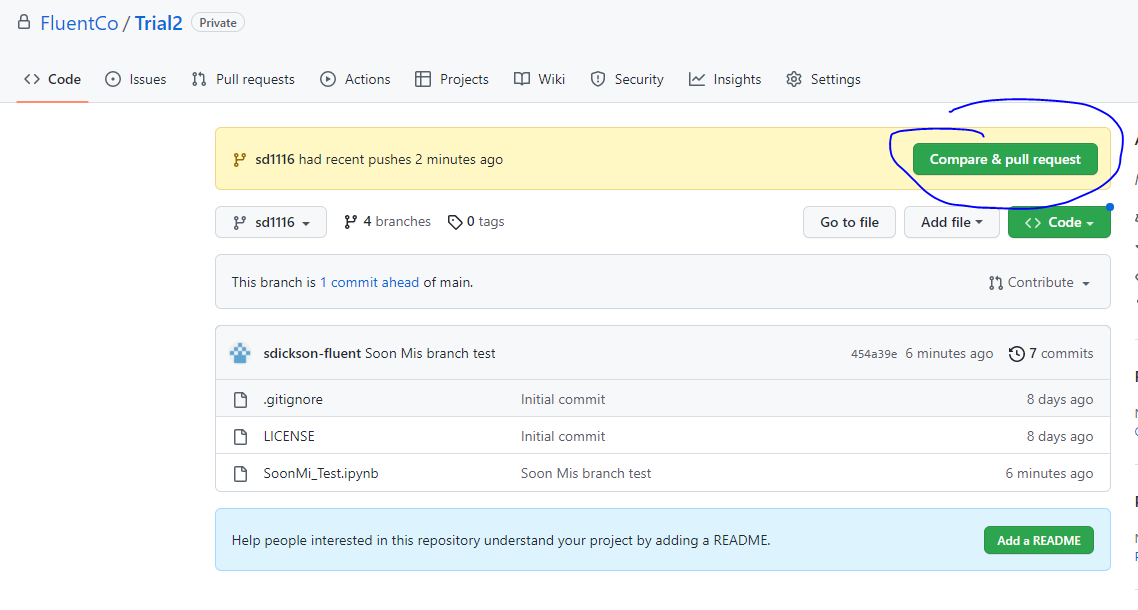


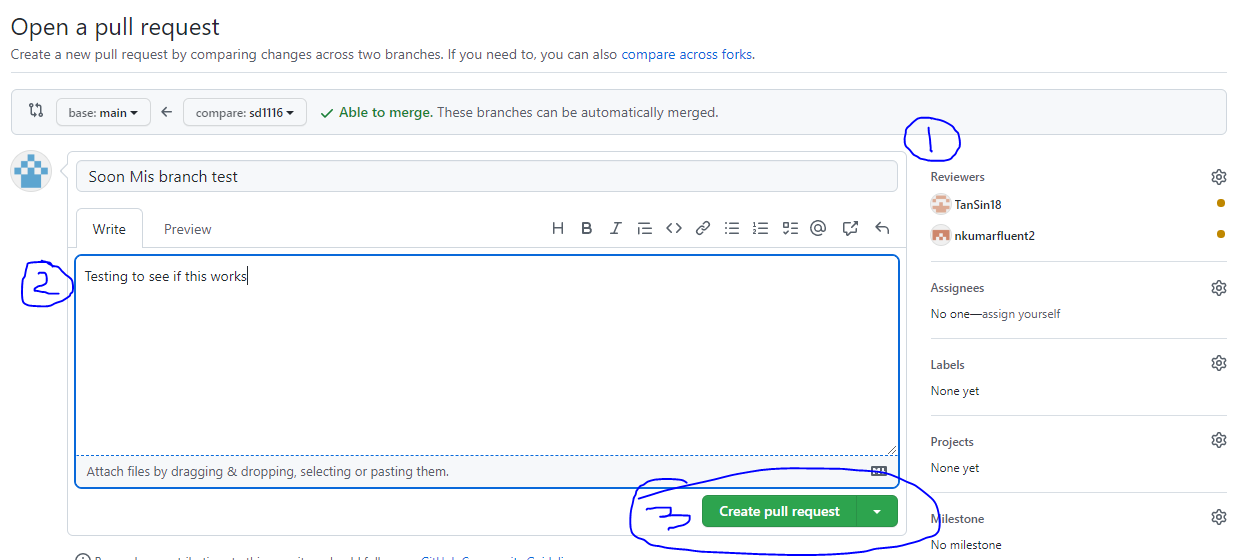
1. Push change upstream

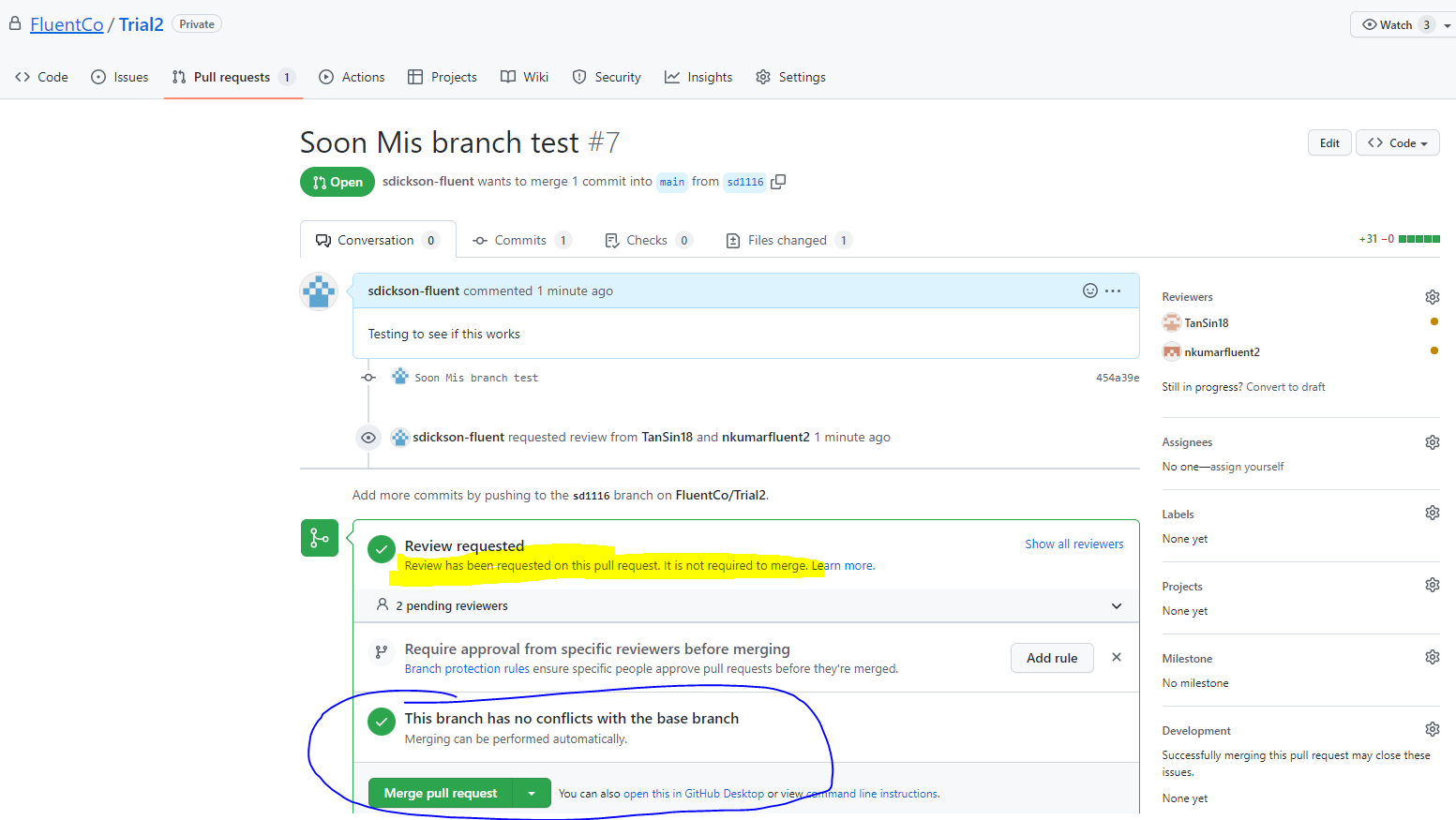
git push -–set-upstream origin “branch name”



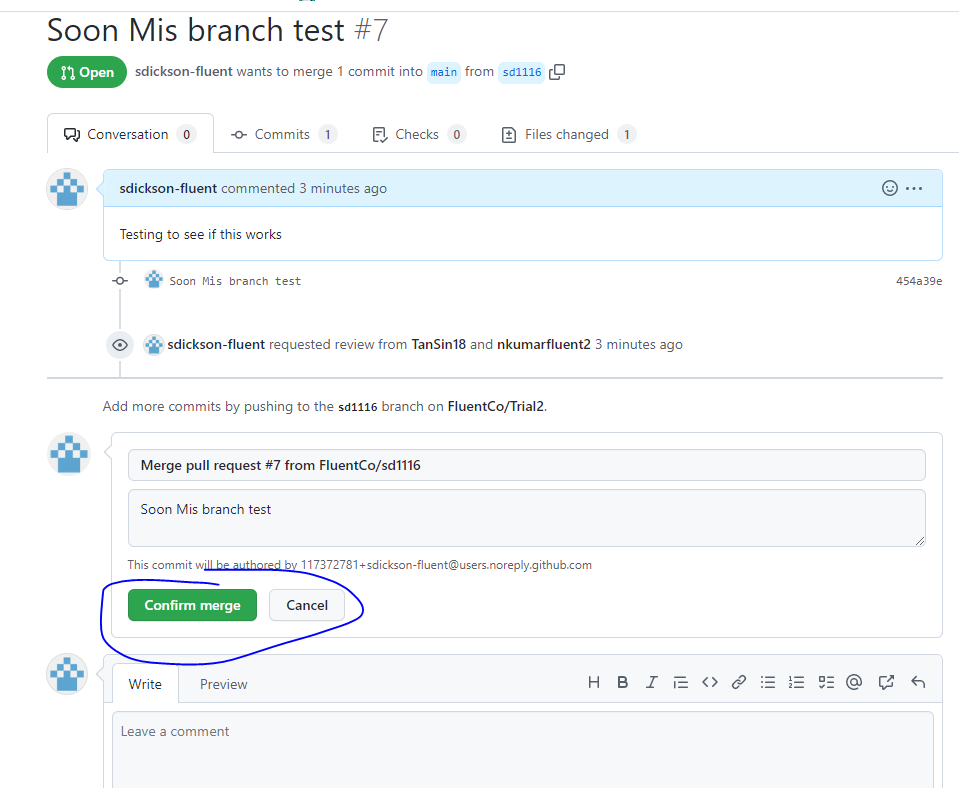
1. Click on “Compare and Pull Request” on Github



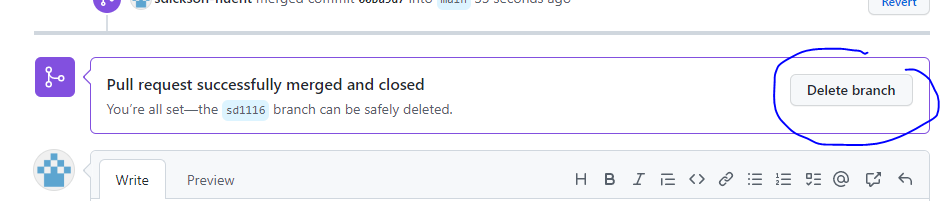
1. Add reviewers and a comment if necessary, then “Create pull request”
2. Reviews can be requested, but are not required. If there are no conflicts, “Merge pull request”



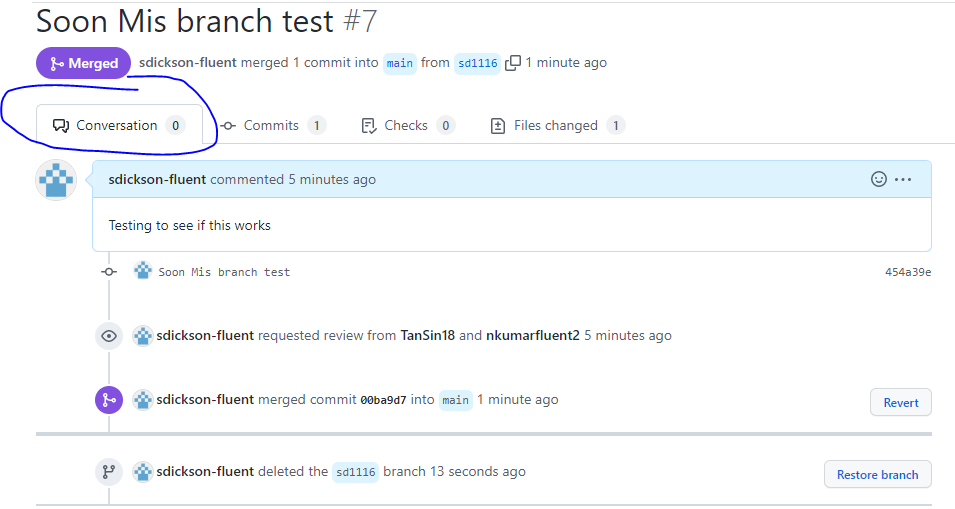
1. Confirm merge. Now the change is reflected in the main branch



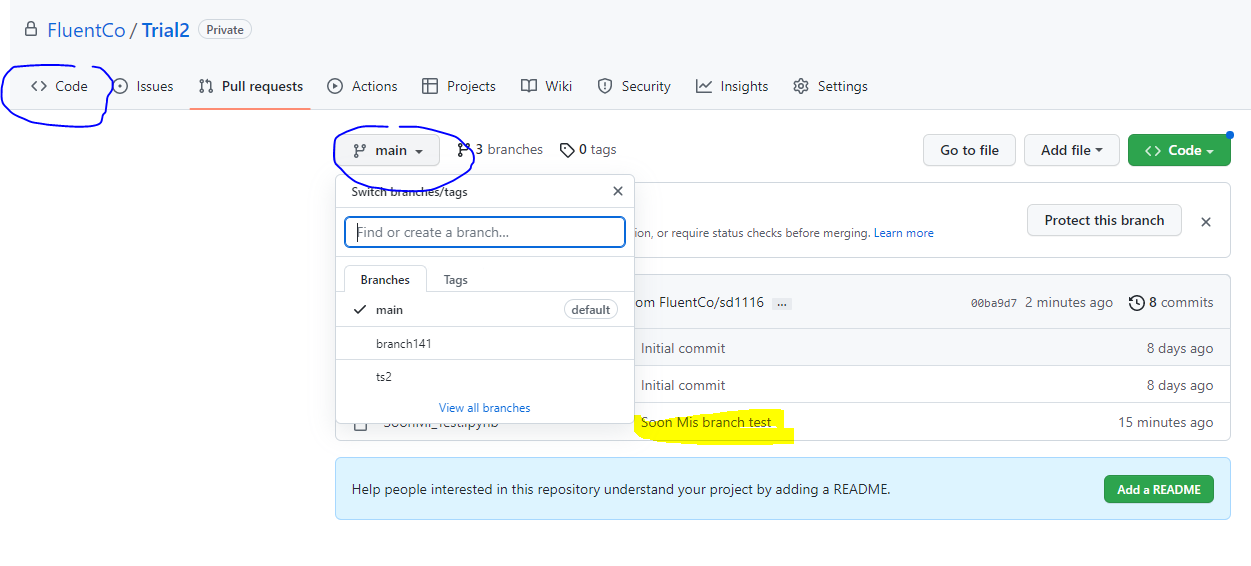
1. Delete branch since it is no longer used



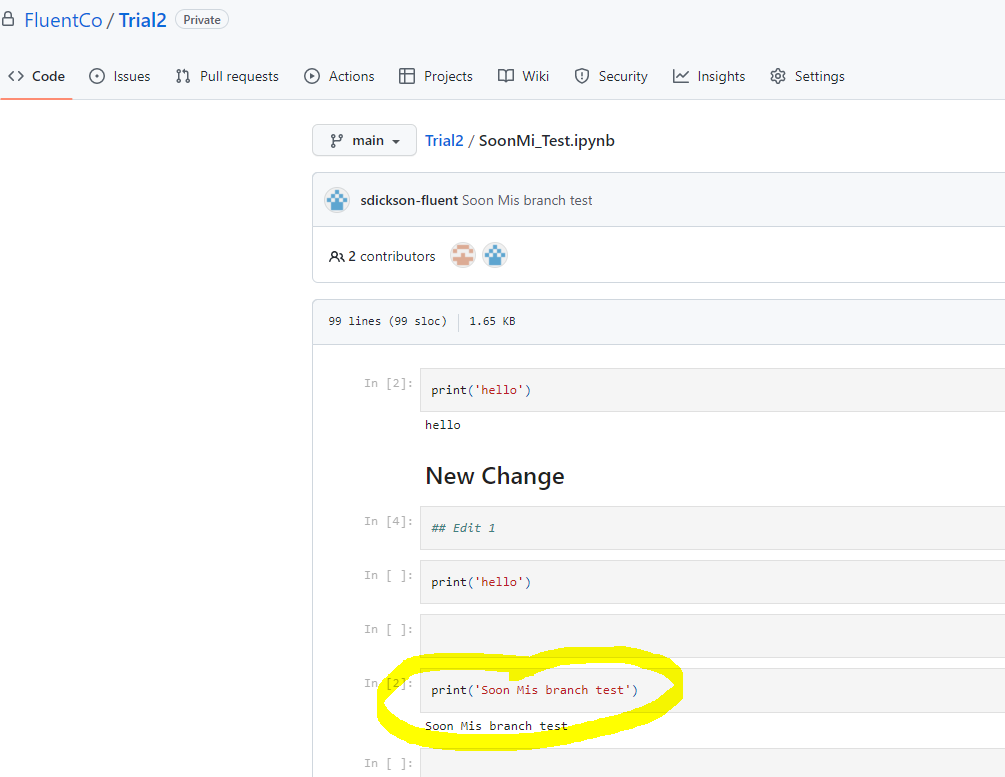
1. The history log is in “Conversation”. You can restore branch if necessary



1. When you go to the Code tab, you can see that the new branch is no longer there. The change you made is now in the main branch



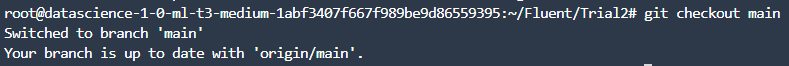
1. When you click into the notebook, the change is there



**Fetching new changes from a repository after you have already cloned it**

1. Check to see if your main branch is up to date

git checkout main



1. If it is not then fetch the updated code from main

git fetch

1. Create a new branch from there