

# GitHub Introduction

Anzony Quispe Rojas

November 15, 2021

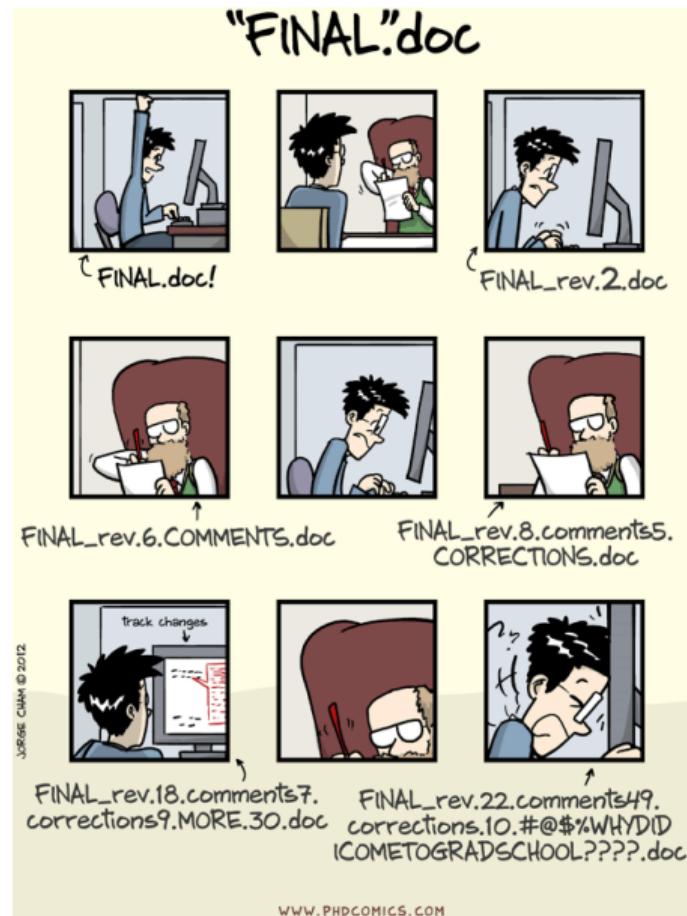
## Citation

These notes are based on the Lecture Notes of DIME Analytics - World Bank.

## Before Starting the class

- Do you have a GitHub account? If not, [click here](#) to sign up.
- Have you submitted your GitHub username to the host  
Anzony Quispe Rojas?
- Have you installed GitHub Desktop? If not, [click here](#) and download it.
- Have you logged into GitHub Desktop at least once? If not, open GitHub Desktop and sign in with your GitHub account.
- Have you been invited to the main repository? And have you accepted the invitation?

# Version Control Issues



## Typical Issues

- Several copies of the same document.
- Problems in tracking changes.
- Accidental mistakes.
- Backups problems.
- Messy documents.
- Team working problems:
  - For Dropbox users, "Copy Conflict" may sounds familiar.
  - Who did change the code?
- No version history.

## Bad Alternative Solutions

- Name files using dates in filenames.
- Name files with version number.
- Email Threading.

## Git - The Best Solution

Git is a version control system developed by Linus Torvalds, the famous creator of the Linux operating system kernel. We can track changes in our files in a highly structured way.

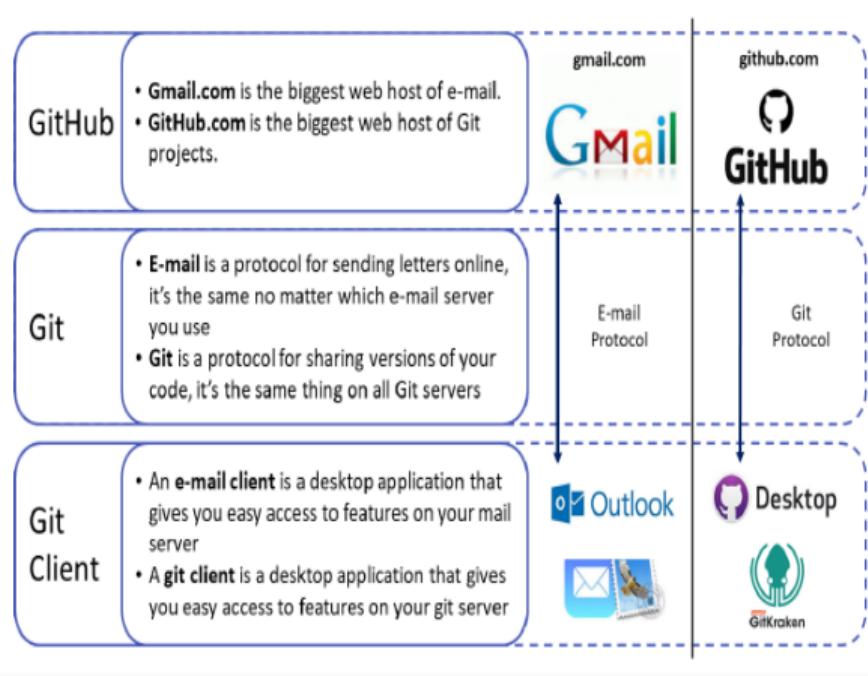
Benefits:

- A complete long-term change history of every file.
- In this workflow, we can work in teams. Anyone can edit; anyone can access previous versions and the most recent versions.
- It makes it easy to get excellent documentation of your project versions.
- Being able to trace each change made and the responsibility for that change.

# GitHub

GitHub is a hosting service. It provides a "home" for your Git-based projects on the internet.

- Showcase your work. Your projects are exposed on the internet and are easily shareable.
- You can easily track the development of any project.
- Everything is in the hosting service. Everything is going to be alright.



## Key Concepts

- **Repository** : Set of files. It exists as a remote repository.
- **Clone** : It is a local copy of all the repository on our computer at that point in time.
- **Commit** : Snapshot of your repository. Git does not store data as a series of changesets or differences but instead as a series of **snapshots**.
- **Branch** : It is simply a lightweight movable pointer to one of these snapshots or commits. A independent line of development.
- **Pull Request** : It is an event when a contributor asks a maintainer of a repository to review code they want to merge into a project.

# Repository

Your project folder is called **repository** in GitHub.

Overview    Repositories 16    Projects    Packages

Find a repository...    Type ▾    Language ▾    Sort ▾

---

**ECO224** Public

Repositorio del curso PUCP Inferencia Causal y Machine Learning

● Jupyter Notebook    ⭐ 3    📈 10    1 issue needs help    Updated 17 hours ago

**ml\_book** Public

Updated 5 days ago

Star



Star



## Clone

It is not the same as downloading. When we clone a repository, we copy the complete history of that repository. Your computer uses Git software to download the version history. Additionally, we can make updates in the cloned repository that can be merged with the remote repository. When we download a repository, we do not copy the project history.

# How to clone?

- Go to the **main page**
- Click on green button **Code** or **download button**.

The screenshot shows a GitHub repository interface. At the top, there are navigation links: 'main' (with a dropdown arrow), '70 branches' (with a dropdown arrow), '0 tags' (with a dropdown arrow), 'Go to file', 'Add file', and a prominent green 'Code' button. Below this, a table lists recent activity:

alexanderquispe Merge pull request #366 from alexanderquispe/ddml	1fd43e8 17 hours ago	853 commits
.ipynb_checkpoints	#1 The first tutorial	3 months ago
Installation	#2 Solving png problems	2 months ago
Labs	#365	17 hours ago

## Commit

It is a snapshot of your repository. Git does not store data as a series of changesets or differences but instead as a series of **snapshots**. It has the metadata: author, date, message, parent commit, tree object hash, parent commit hash, and message. It indicates a significant modification to our version control system. Finally, a commit object references a snapshot.

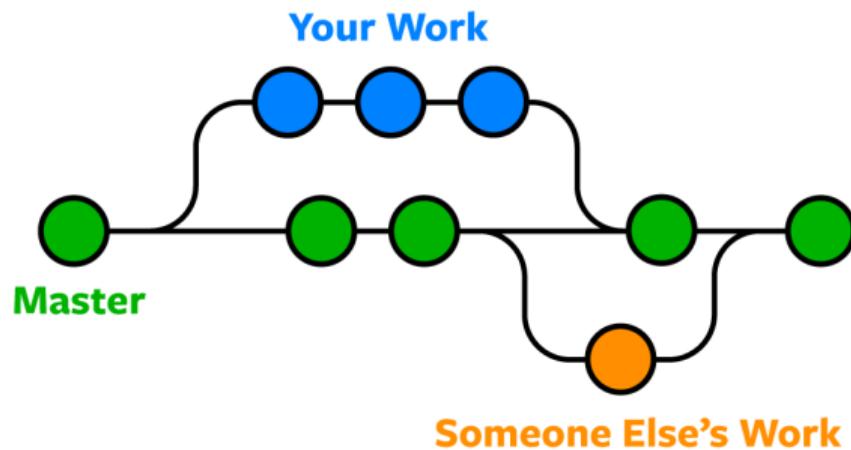
Each commit has an edit date and trace of who made that significant change.

Before we make a commit, we need to learn what is a branch.

We need to practice to learn. We are going to create a new file and store it in a specific folder.

# Branch

Using Branches is Git's "Killer" tool. This feature of Git is what makes it so powerful as a collaboration and version control tool. It allows you to refer to any snapshot of your project.

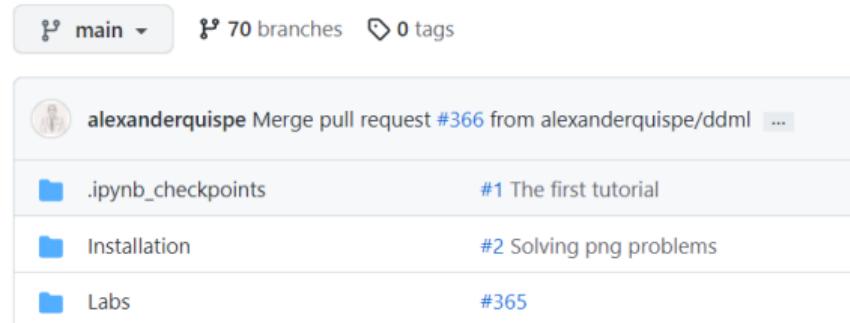


## Key questions about branches

- What happens to the contents of the folder on your computer when you check another branch in GitHub Desktop?
- What is the version that is cloned on your computer? Actually, all are cloned, but only one is shown, that branch you work on.

# Create a Branch

- Go to our repository in GitHub.



- Write your name in that field and click on Create branch: `your_name`. Make sure it says "from master".
- Now another branch should appear with `your_name`
- Go to this link to see your branch.

## Pull Request

PULL REQUEST = Request that your modifications be transferred to the **master branch**.

Usually, there is always a person who is called the **REPO MAINTAINER**, who has access to the **master branch**. Therefore, the only way to contribute to the **master branch** is through a **pull request**.

# How to do a Pull Request

- Go to the **Pull Request** section of our repository.
- Click on **New Pull Request**
- Select your branch
- Check if the modifications you are requesting for the master branch. **This step is important because in case of any conflict, you will not be able to unite your changes.** If everything is ok, then click on **Create Pull Request**.
- Finally click on **Create pull request** again.

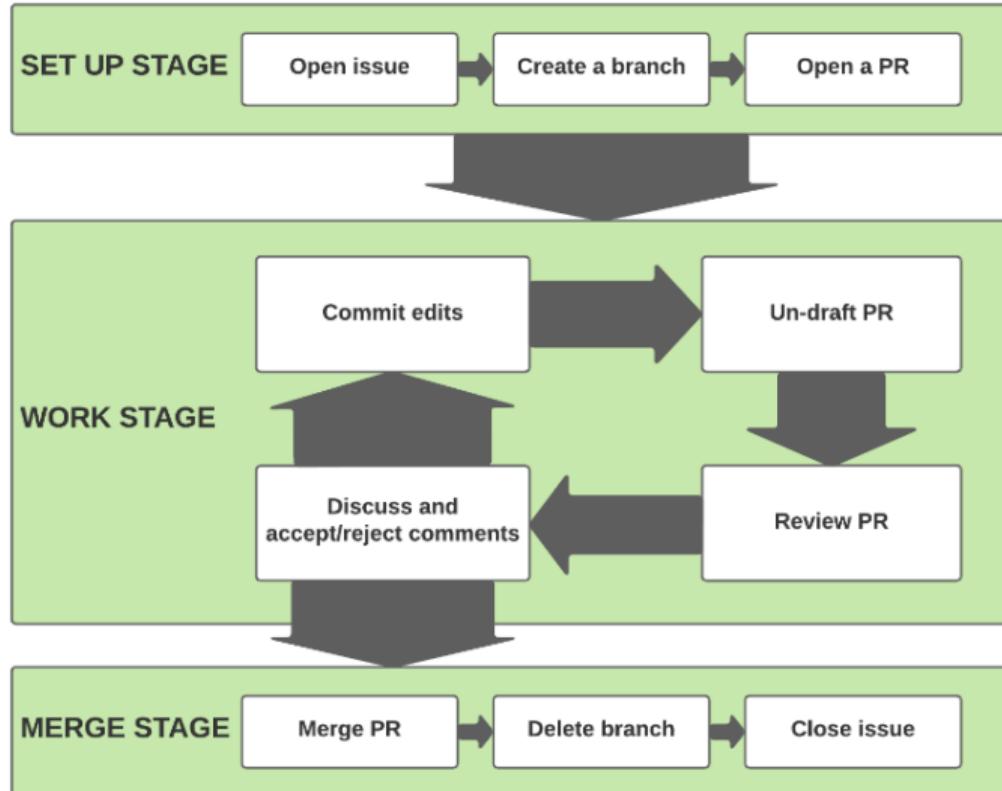
## Branch-PR-merge Cycle

This is the best practice framework for implementing each task in a Pull Request.

It has three stages. In this course, all the students must complete a full branch-PR-merge cycle for all the tasks. It will be part of the evaluation.

- Set up Stage
- Work Stage
- Merge Stage

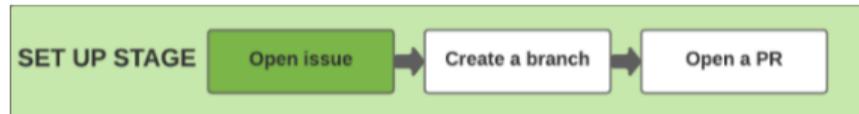
# Branch-PR-merge Cycle



## Set up Stage

This stage has three main sections.

- Open issue
- Create a branch
- Open a PR.



# Issues

We create a space in our repository to work on. This space is named as **Issue**.

- The issues allow us to document every significant change in our repository.
- We can assign and classify tasks.
- Team members can provide insights on how to solve a specific task.

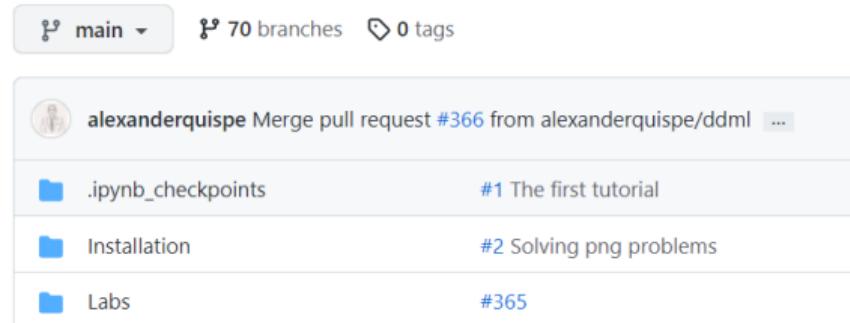
<input type="checkbox"/> <span style="color: green;">●</span> Days to withdrawal <span style="background-color: #f0f0f0; border: 1px solid black; padding: 2px;">question</span>	#251 opened on Dec 15, 2020 by keybormy			2
<input type="checkbox"/> <span style="color: green;">●</span> Code review	#259 opened on Dec 15, 2020 by keybormy			3
<input type="checkbox"/> <span style="color: green;">●</span> Decomposition extensive / intensive margin	#254 opened on Dec 15, 2020 by keybormy			1
<input type="checkbox"/> <span style="color: green;">●</span> Urban vs. rural <span style="background-color: #f0f0f0; border: 1px solid black; padding: 2px;">priority: high</span>	#250 opened on Dec 15, 2020 by keybormy			1
<input type="checkbox"/> <span style="color: green;">●</span> Combination variable - failure to withdraw and/or zero withdrawal recorded in tranche	<span style="background-color: #f0f0f0; border: 1px solid black; padding: 2px;">presentations</span> <span style="background-color: #f0f0f0; border: 1px solid black; padding: 2px;">severity: high</span>	#251 opened on Dec 14, 2020 by keybormy		6
<input type="checkbox"/> <span style="color: green;">●</span> Back of envelope estimate of cost of biometric ATMs (and ATM coverage overall)	#250 opened on Dec 14, 2020 by keybormy			
<input type="checkbox"/> <span style="color: green;">●</span> Gross vs. net interpretation	#255 opened on Dec 11, 2020 by keybormy			
<input type="checkbox"/> <span style="color: green;">●</span> Decomposing first stage <span style="background-color: #f0f0f0; border: 1px solid black; padding: 2px;">severity: medium</span>	#252 opened on Dec 11, 2020 by keybormy			5
<input type="checkbox"/> <span style="color: green;">●</span> Geocode all OPM HHs at the village level <span style="background-color: #f0f0f0; border: 1px solid black; padding: 2px;">severity: medium</span>	#278 opened on Nov 11, 2020 by amergal94			3
<input type="checkbox"/> <span style="color: green;">●</span> Payment point proximity	#277 opened on Nov 4, 2020 by keybormy			0 of 2
<input type="checkbox"/> <span style="color: green;">●</span> Compare zero withdrawals in overall MIS vs. in matched sample w/OPM	#279 opened on Nov 4, 2020 by keybormy			
<input type="checkbox"/> <span style="color: green;">●</span> Shift into & out of zero withdrawals (breaking down extensive margin effect)	#275 opened on Oct 26, 2020 by keybormy			
<input type="checkbox"/> <span style="color: green;">●</span> Specifications with payment point data <span style="background-color: #f0f0f0; border: 1px solid black; padding: 2px;">OPM &amp; MIS</span> <span style="background-color: #f0f0f0; border: 1px solid black; padding: 2px;">severity: high</span>	#274 opened on Oct 21, 2020 by keybormy			2

## How to create an Issue?

- Click on issue tab **New Issue**.
- Add descriptions and assign it properly.
- Add labels to facilitate future searches.

# Create a Branch

- Go to our repository in GitHub.



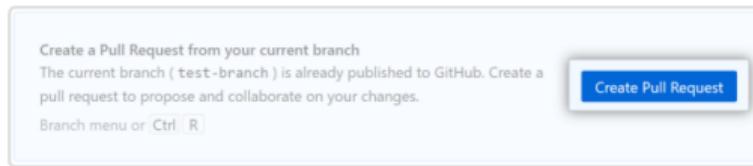
- Write your name in that field and click on Create branch: `your_name`. Make sure it says "from master".
- Now another branch should appear with `your_name`.
- Go to this link to see your branch.

## Open a Pull Request

- It creates a space on GitHub where the progress of this task can be followed.
- We create a *draft PR* that cannot be merged while in draft status.

# How to create a Pull Request

- Switch to the branch that you want to create a pull request for.
- Go to *Pull Request* tab.



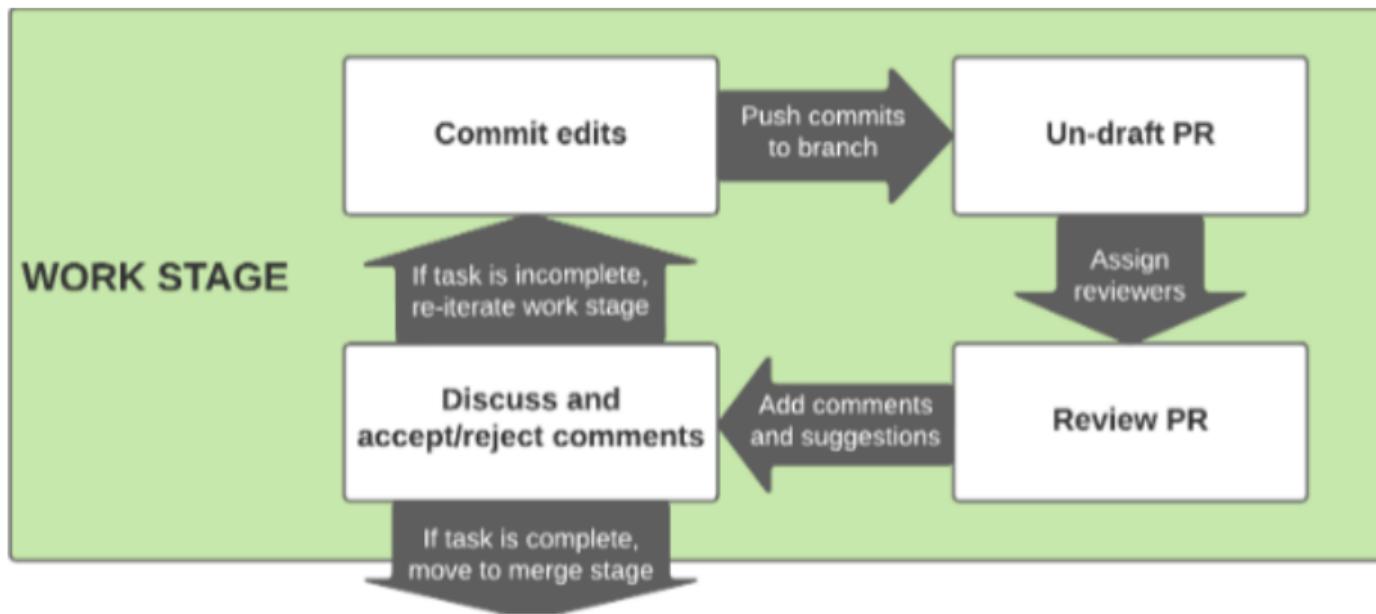
- Please, confirm that the branch in the base is the branch where you want to merge your changes.
- Name your **Pull Request**.
- We can label our **Pull Request**.

## Work Stage

Complete your tasks.

- Commit edits: Commit every significant change.
- The Pull Request will get every commit we made.
- Make commits until the task is completed.

# Work Stage



# Work Stage

- Un-draft Pull Request: Assign reviewers in case you work with a Repository Maintainer (RM).
- You can tag team members to be part of the discussion.

The screenshot shows a GitHub pull request interface. At the top, there's a code diff section with several commit messages. One commit message is highlighted in green and contains the following text:

```
00 -27-18 +27,15 @@ La version que ustedes **"Undran** al master branch no será la que envías cuando
27 27
28 28 Sugerencias :
29 29 1. Dar un nombre al PR un nombre asociado al nombre del branch, pero un poco más descriptivo.
30 30 2. Pueden agregar un label _"Work in Progress"_.
31 31 - 3.
32 32 + 3. Solo cuando hayan terminado de hacer todos los **commits** necesarios deberian solicitar o realizar el merge.
33 33
34 34 # Work Stage
35 35 + 1. El autor hace los **commits** como siempre.
36 36 + 2. El PR se actualizara con cada **commit** que ustedes realicen.
37 37 + 3. Sigue agregando **commits** hasta que hayan terminado la tarea.
38 38 +
39 39 +
40 40 +
41 41
```

Below the code diff, there's a comment section. A comment from user `alexanderguispe` is shown, dated 19 minutes ago:

alexanderguispe commented on `se6309e` 19 minutes ago

Collaborator Author ⚙️ ...

Creo que deberías revisar nuevamente las líneas de código que has agregado.

At the bottom, there's a compose area for a new comment:

Write Preview

@antonioal23 deberías revisar estas nuevas líneas que se han agregado.

Attach files by dragging & dropping, selecting or pasting them.

Comment on this commit

## Merge Stage

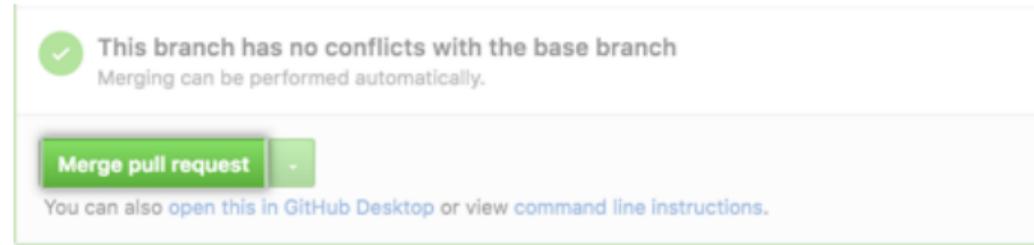
We merge the Pull Request only when we finished with the task.

- Your task must be properly documented.
- Future team members should be able to understand the state of the issue by reading your documentation.
- Delete the branch. We do not want to work on an outdated version of the repository. If you want to work in the same task in the future, you have to delete the branch and create a new one.

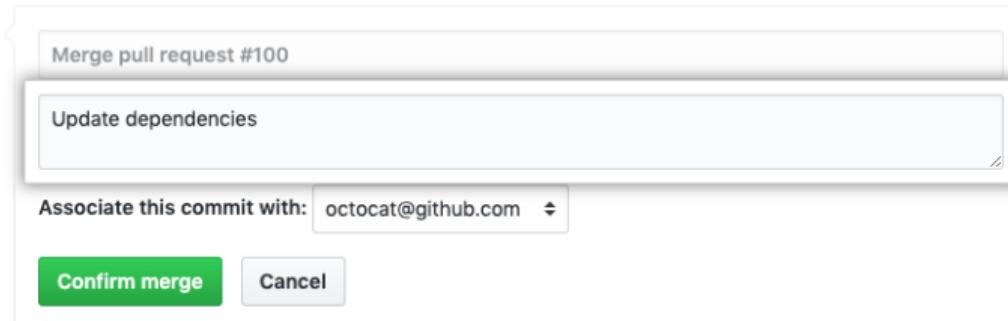


## Merge Stage

Merge your Pull Request.



Confirm your Merge.



Finally, close the issue when the task is done.