# Recommended Procedures for Submitting Robot Code Changes

# Team 3630

# January 10, 2019

## Assumptions

This document assumes that the team member has performed all of the steps in the WPILib manual for setting up a proper development environment here:

<https://wpilib.screenstepslive.com/s/currentCS/m/java/c/88899>

This document also assumes that the team member already has an account on github.com. If you do not yet have an account, one may be created at: <https://github.com/join>

## Initial Setup – One Time

1. **Install Git on your local machine:**

To perform this setup, simply visit <https://git-scm.com/book/en/v2/Getting-Started-Installing-Git> and follow the instructions for your operating system.

1. **Configure Git global variables for your environment:**

Open up your machine’s terminal, and run the following 2 commands:

|  |
| --- |
| git config --global user.email <Email address associated with your github account here> |

|  |
| --- |
| git config --global user.name <The name of your github account here> |

1. **Fork the official team repository to your GitHub account:**

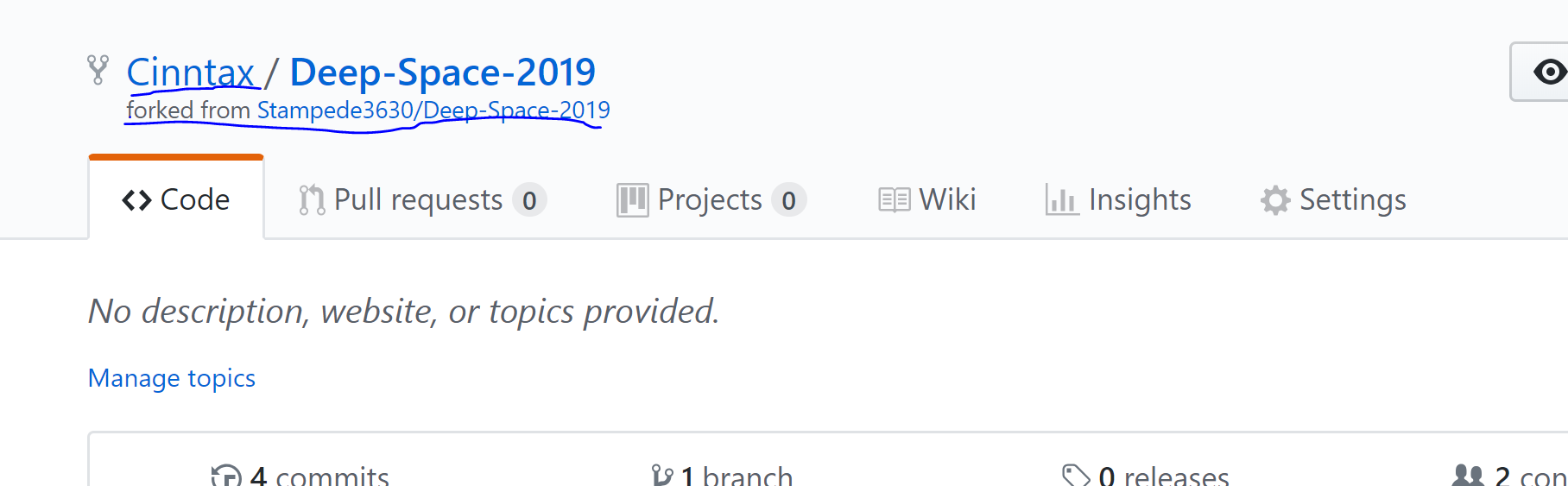
Why?: Git (and GitHub) are meant to be distributed

Login to your github account, and navigate to: <https://github.com/Stampede3630/Deep-Space-2019>

Click on the “Fork” button to create your own version of the repository that you will use to propose your code changes to the team:

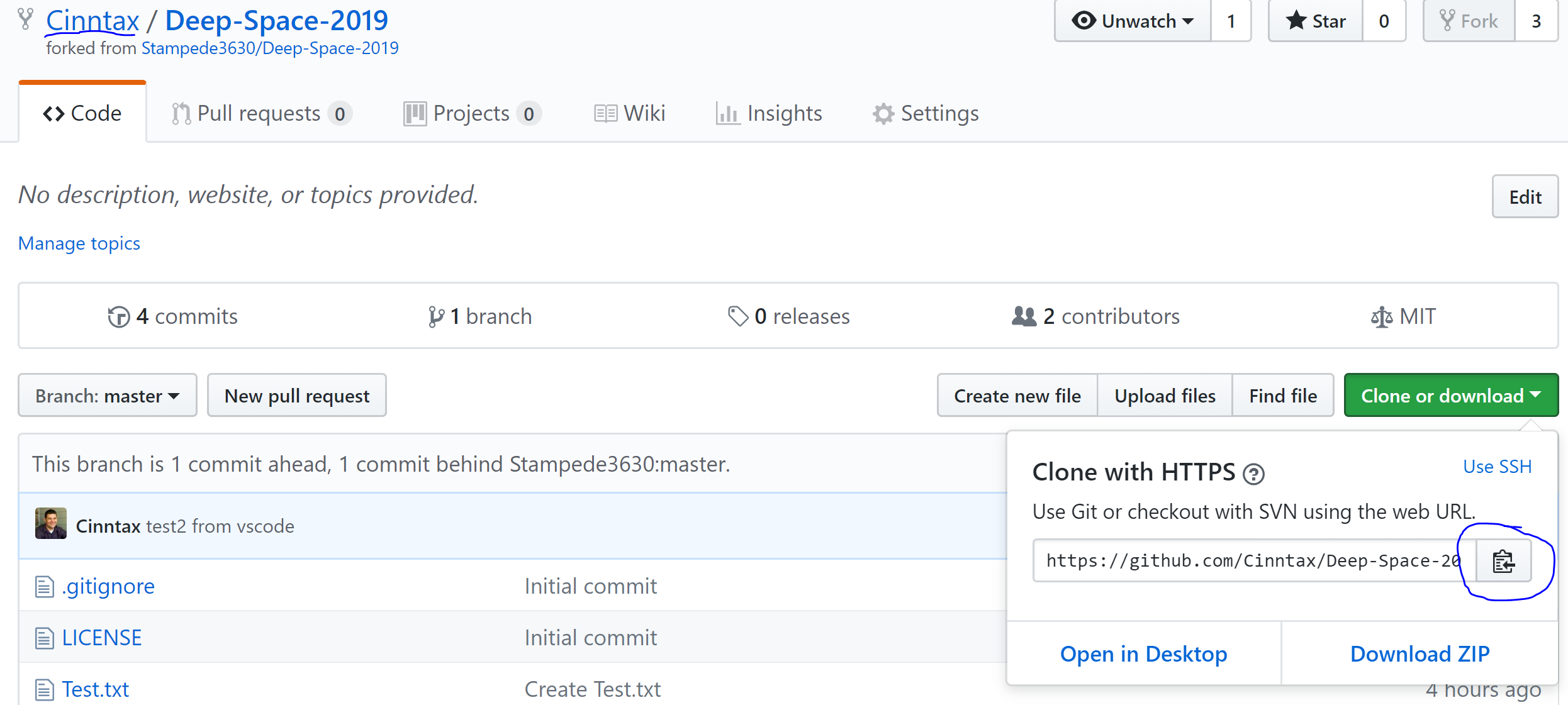


After this step is complete, you should see a screen that looks like this:

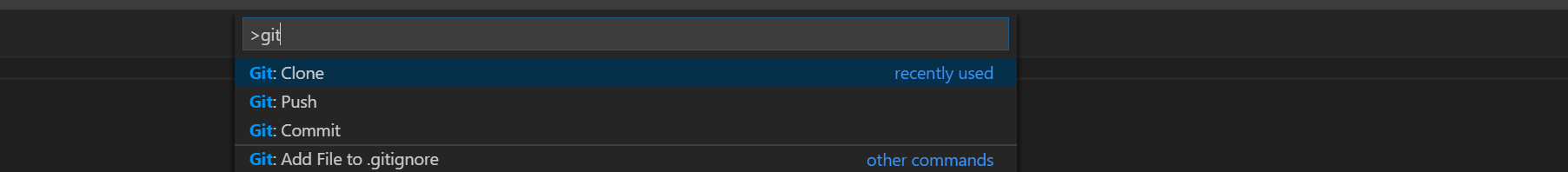


1. **Clone the official team repository to your local machine:**

To perform this step, you’ll need to obtain the git link to your new repository on GitHub. It can be found by clicking on the “Clone or download” button as shown:

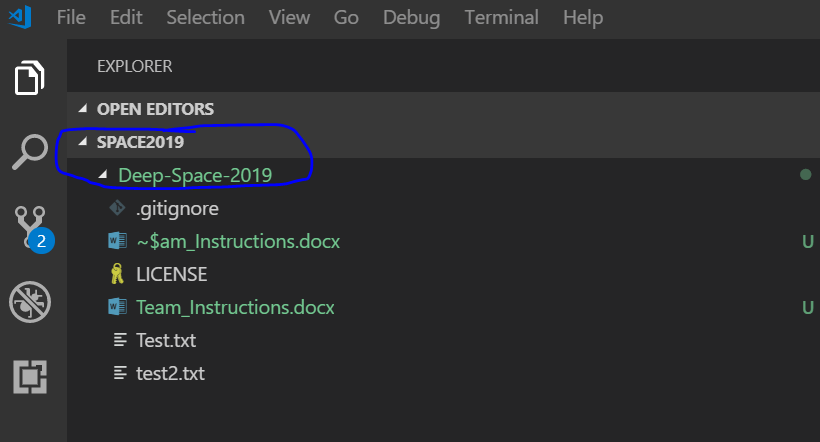


In VSCode, use the “Git Clone” command to pull the content down from your fork of the project.



In the input field shown, paste in the URL copied from the previous step. You will be asked to provide a folder to house the files downloaded. You may select any folder on your machine that you wish. You will also be asked if you wish to “open” this folder afterward. The answer is YES.

After this step is complete, you should have the project’s codebase copied to your machine, and visible within VSCode.

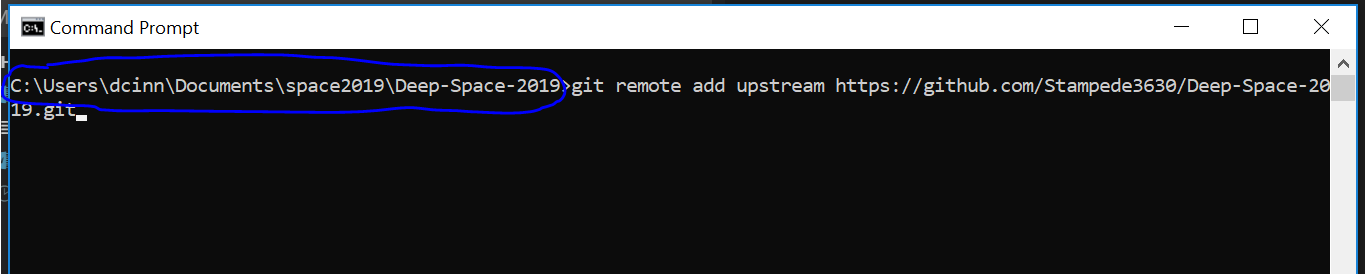


1. **Add a reference to the central repository in your local git repository**

This will allow you to easily/quickly pull the latest/greatest code from the team’s GitHub repository.

Open up your machine’s terminal, make sure you’re in the folder chosen in the previous step and run the following command:

|  |
| --- |
| git remote add upstream https://github.com/Stampede3630/Deep-Space-2019.git |



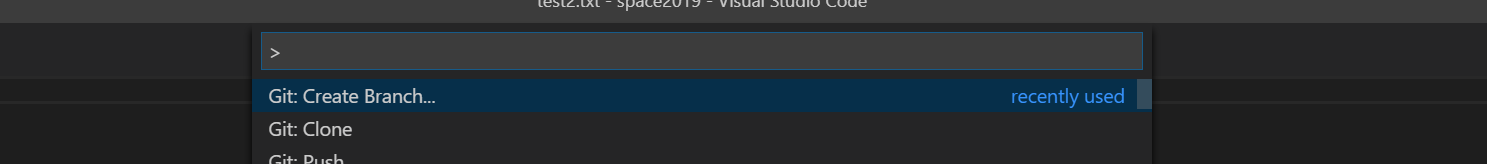
## Submitting a Change – Overall Process

Here is an overview of the process from beginning to end:

1. **Create a new “feature branch” in git**

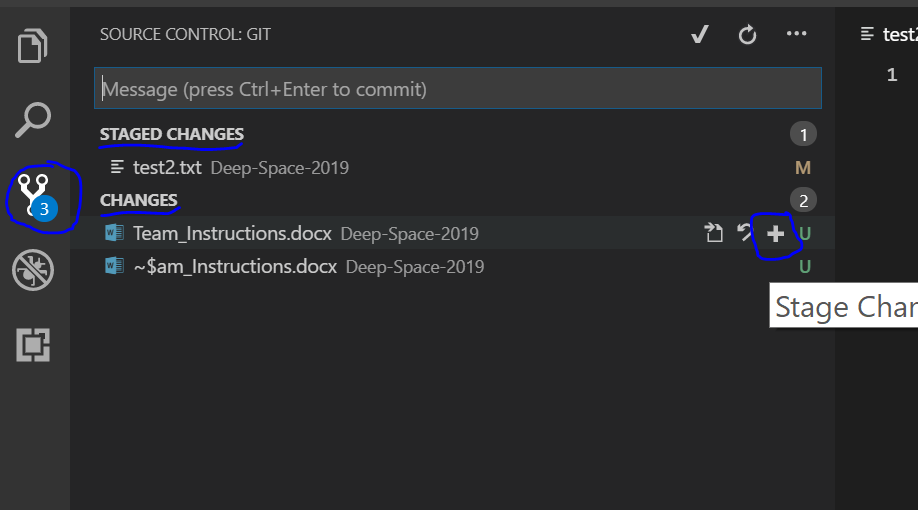
When working on a particular task, it can be helpful to create its own branch within the git repository, and only commit changes related to that task within that branch. This way if you are working on multiple tasks in a short time period, each related set of changes can be managed and submitted to the team independently of each other as they are completed.

In the command palette, run the “git: create branch” method, and give your branch a nice name to reflect the work you’re doing.



1. **“Stage” the changes in git**

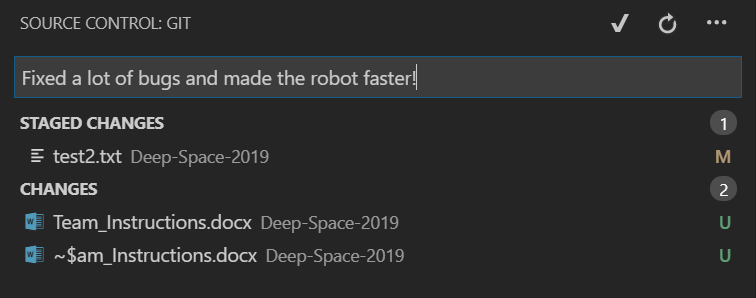
This is a process locally on your machine where you can flag which files/changes you wish to commit into git. This allows you to only commit changes that are ready. If you do not stage a change, that file/change will not make it into git, and ultimately not to the rest of the team.



Using the change control menu (circled) within VSCode, you may control what changes are ready for commission in git. In this example, all files underneath “Staged Changes” will be committed in the next step, and those underneath the “Changes” will NOT be committed.

1. **“Commit” the changes in git**

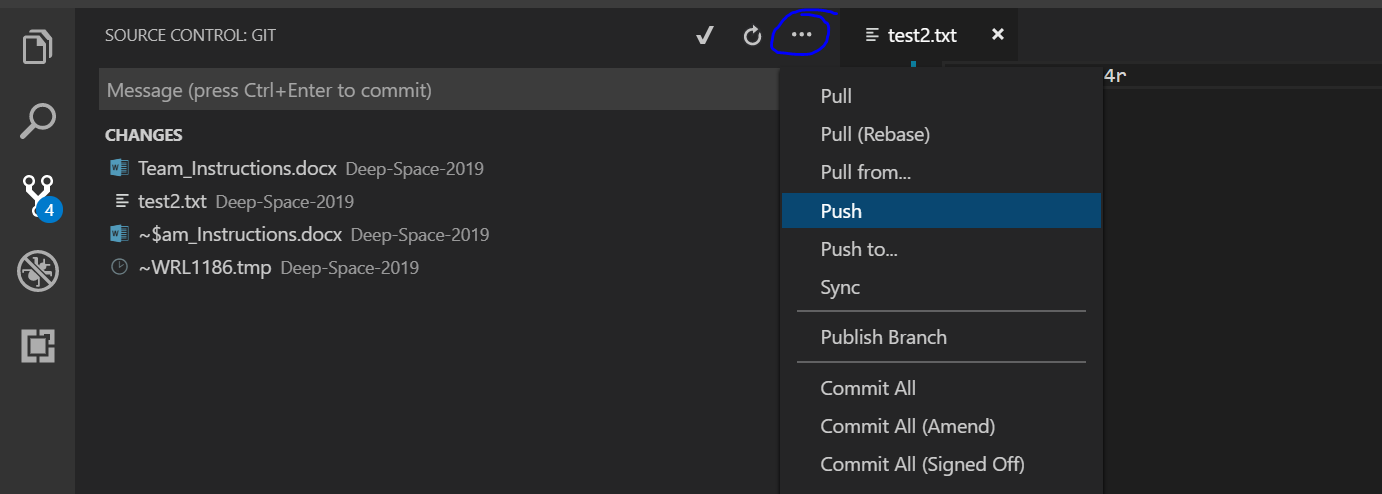
This is a process ***locally*** on your machine where you commit a particular change into the git repository. To commit, simply add a message into the commit box, and press “enter”



1. **Push the changes to GitHub**

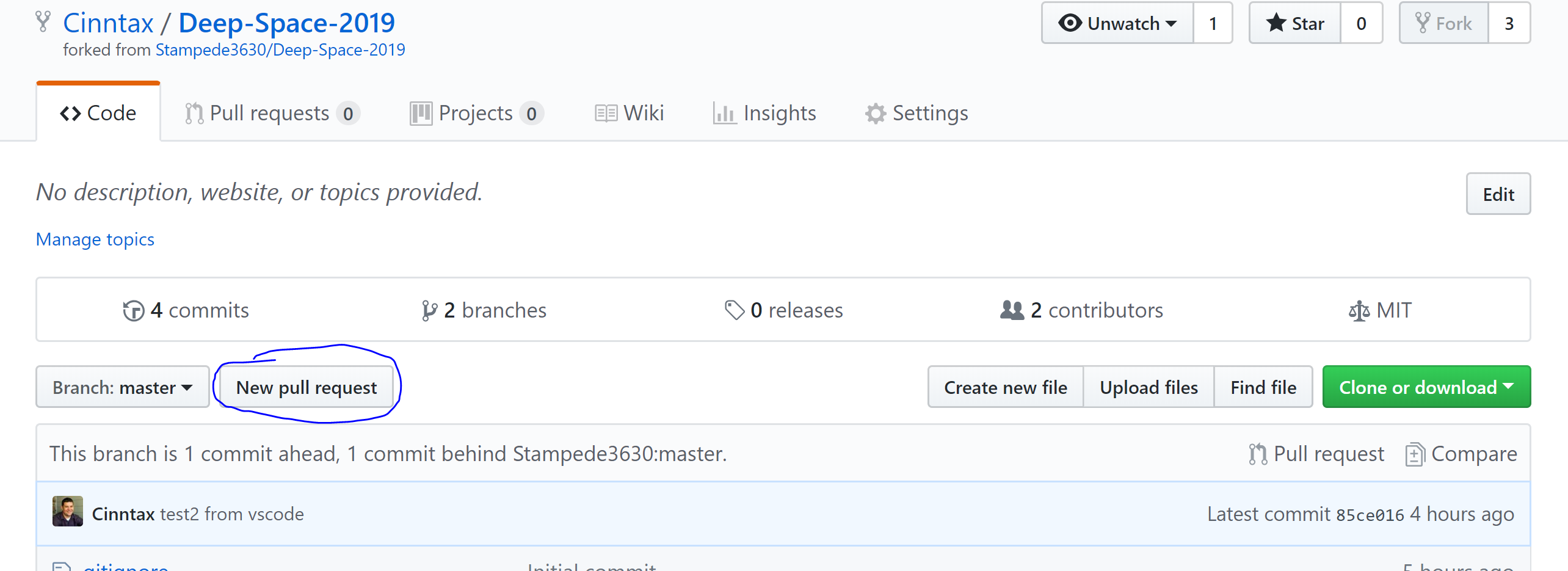
This process will push your code to GitHub, where it may be proposed to the team. This will publish the changes to YOUR fork of the project, not yet the team’s central repository.

You may be asked if you wish to “publish your branch”- the answer is YES.



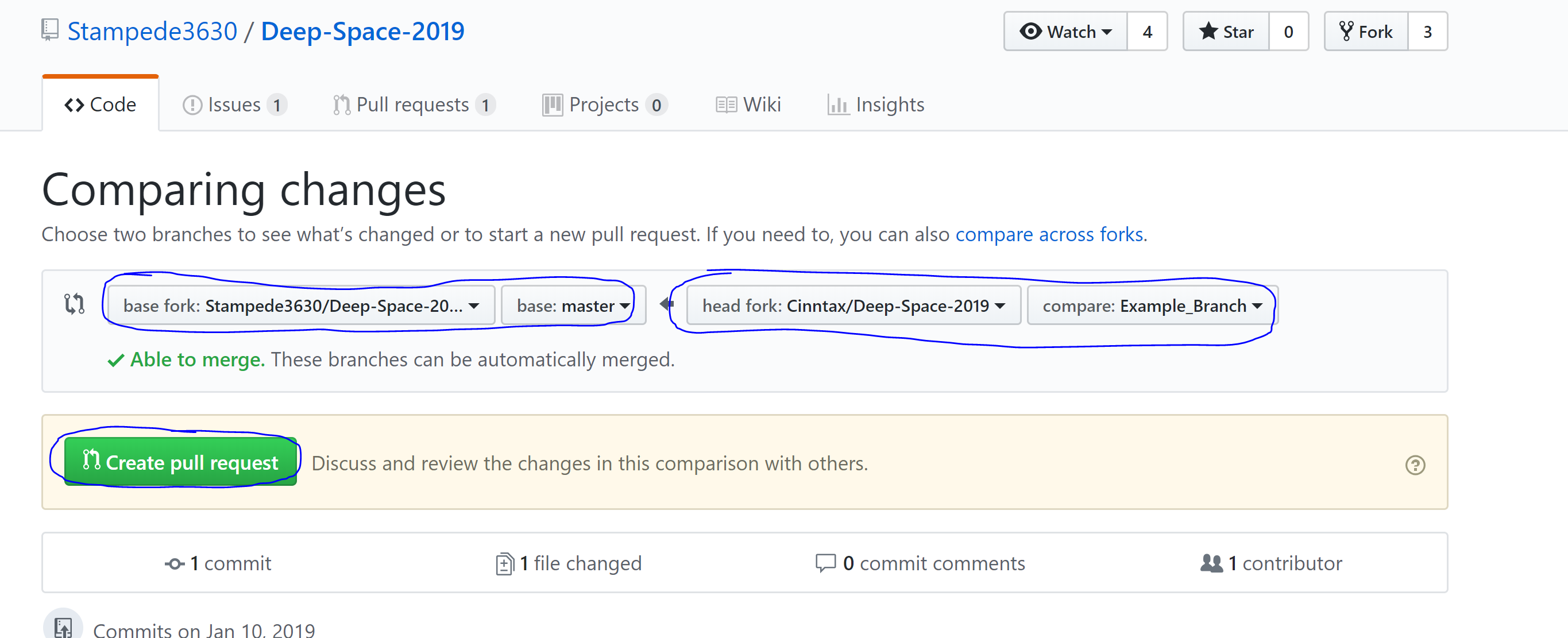
1. **Create a pull request**

This is where you submit your changes to the central team- changes will be reviewed by the team leader prior to acceptance. In GitHub, you should see your latest commit that you just pushed up in the previous step. Use the “New Pull Request” button to submit your changes to the central team.



On the right you should see your repository, and your feature branch- as shown below.

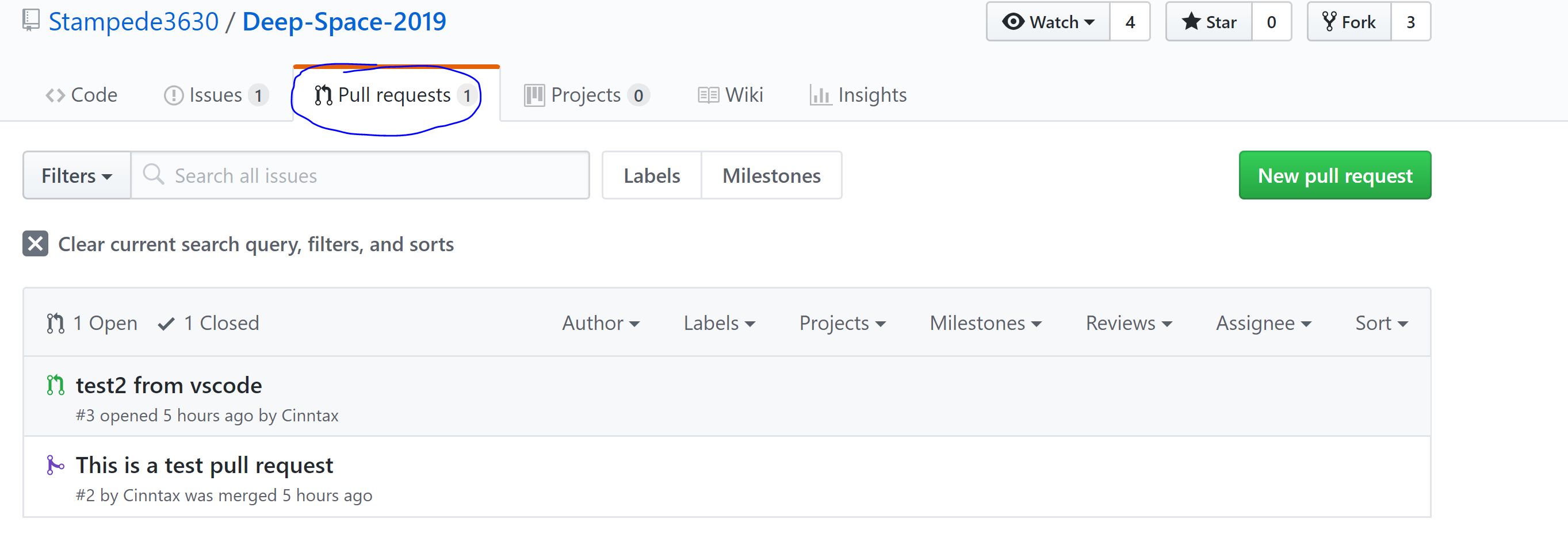
On the left you should see the central respository, and the master branch- as shown below. Press “Create Pull Request” to submit the changes for review.



DONE!

## Checking the Status of Your Request

To see if your change has been accepted or not, go to the “pull requests” tab on the central repository page here: <https://github.com/Stampede3630/Deep-Space-2019/pulls?utf8=%E2%9C%93&q>=



## Yay- my change was accepted. Now What?

A few things:

1. Your “feature branch” is no longer necessary- it should be deleted. In VSCode, run the “Git: Delete Branch” command. In GitHub, there is also a button to delete the branch in GitHub.
2. You’ll want to refresh your git repository from the central repository to get all the changes that your teammates have done! To do this, do a “pull” from the “upstream” remote. This will cause git to go out to the CENTRAL repository, get all changes, and automatically merge them in locally. Now you may start all over at step 1 again with a new feature branch. That feature branch will use the latest/greatest central repository code as a starting point.

