Research GitHub Work Flow

Andrew Leach

Summer 2019

Introduction to version control

Team goals

What is Git?

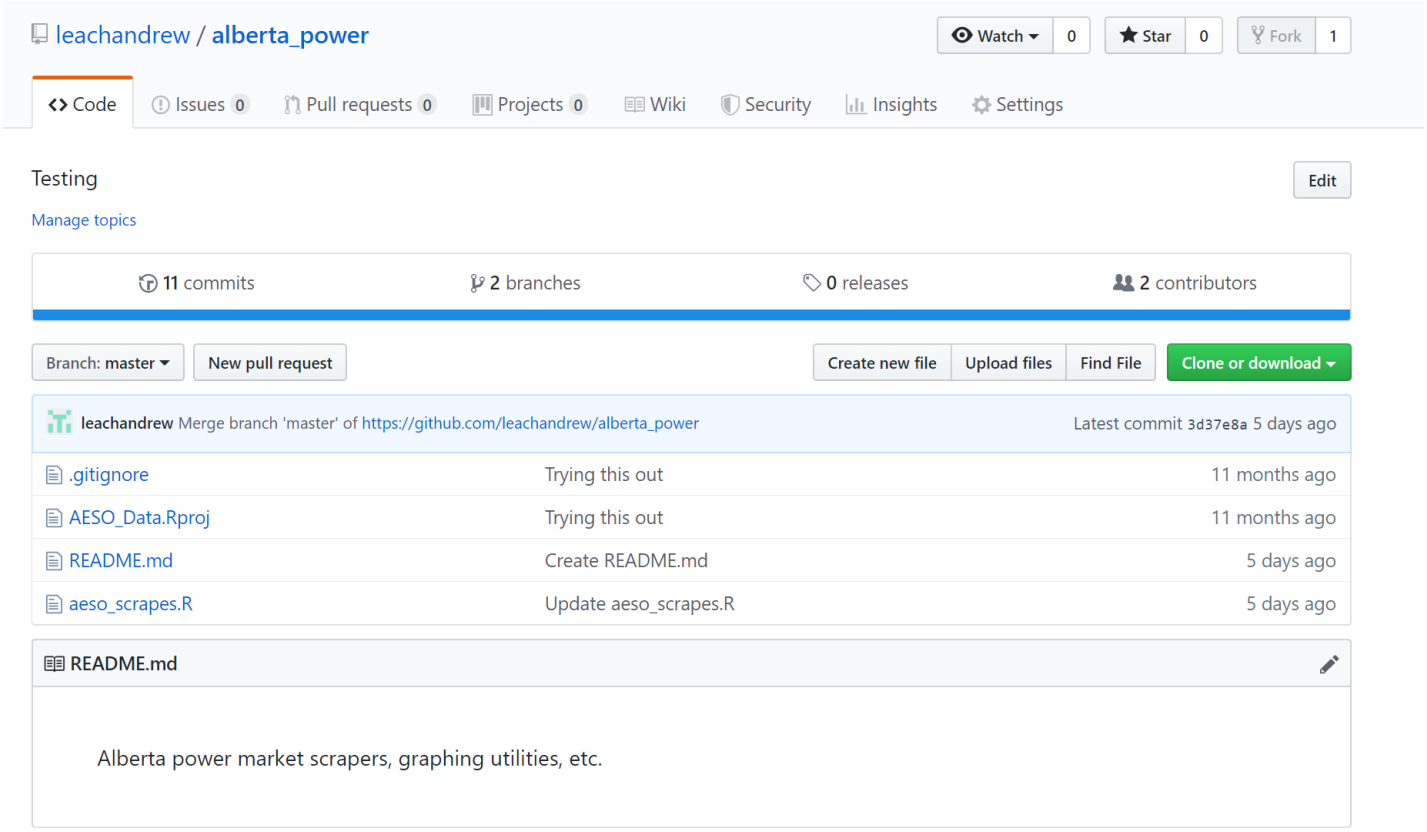
Git is open-source version control software. Using Git, you can create local and remote (cloud) repositories for your files. Git will allow you to track and label updates to those files (kind of like Google Drive) with comments so you know what you've done. You can also use Git (and interfaces like GitHub) to contribute to other people's code and/or allow other people to modify your code.

How are we going to use it?

Basically, I'll maintain the code archive. I encourage you to use my code and, as you make changes you think should be implemented for the group, it would be helpful if you would push them back to me. Below, I've highlighted two example workflows by which we can make this work.

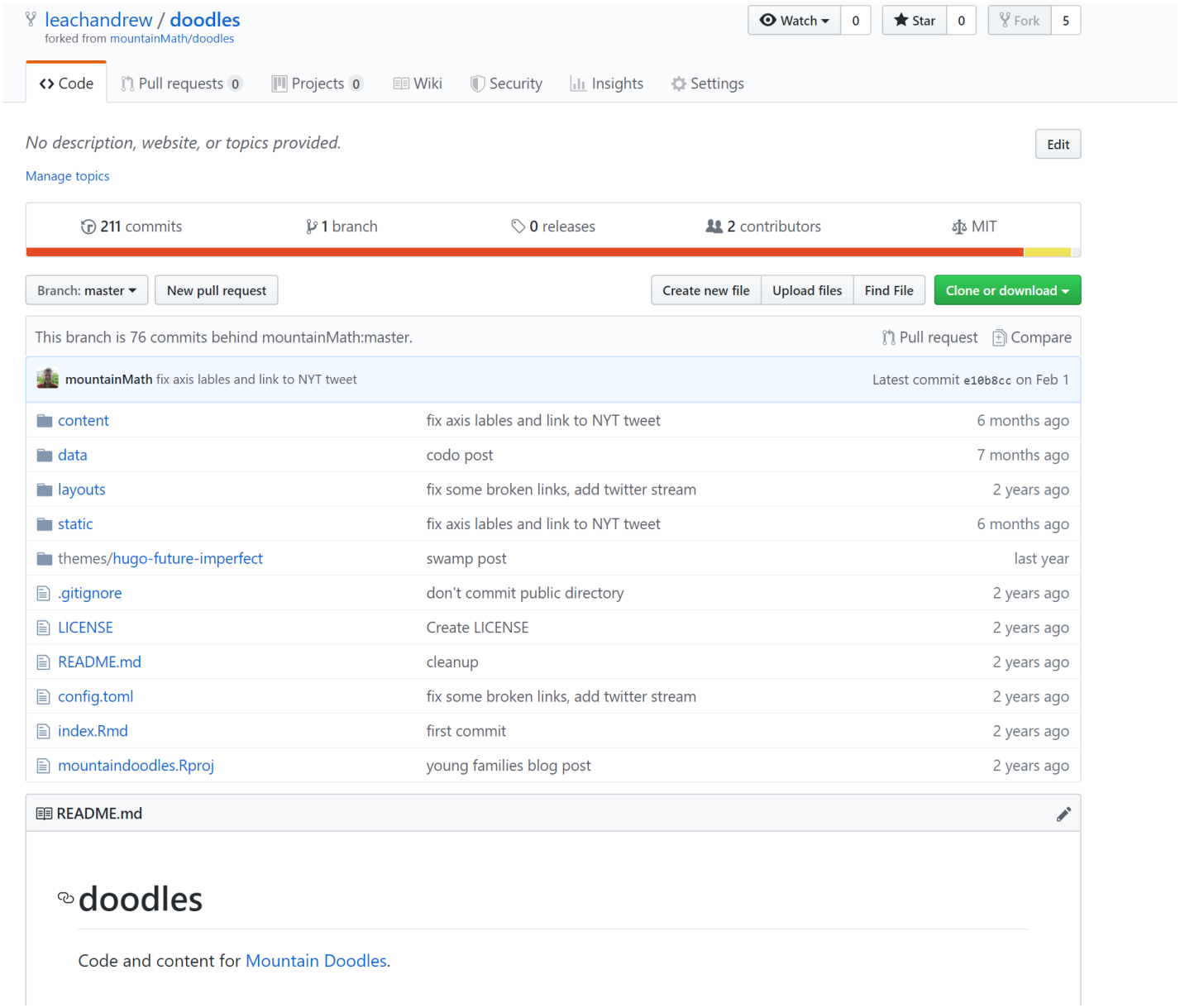
GitHub -> Fork –> Your computer –> Edit –> Commit changes –> Pull request back to me

This is the most basic GitHub-based workflow, and will require you to, at a minimum, create a GitHub account. Once you've got a GitHub account, go on to GitHub (https://github.com/leachandrew) and find the repository you want. For example, if you're looking for AESO scrapers, you could choose the alberta_power (https://github.com/leachandrew/alberta_power) repository.

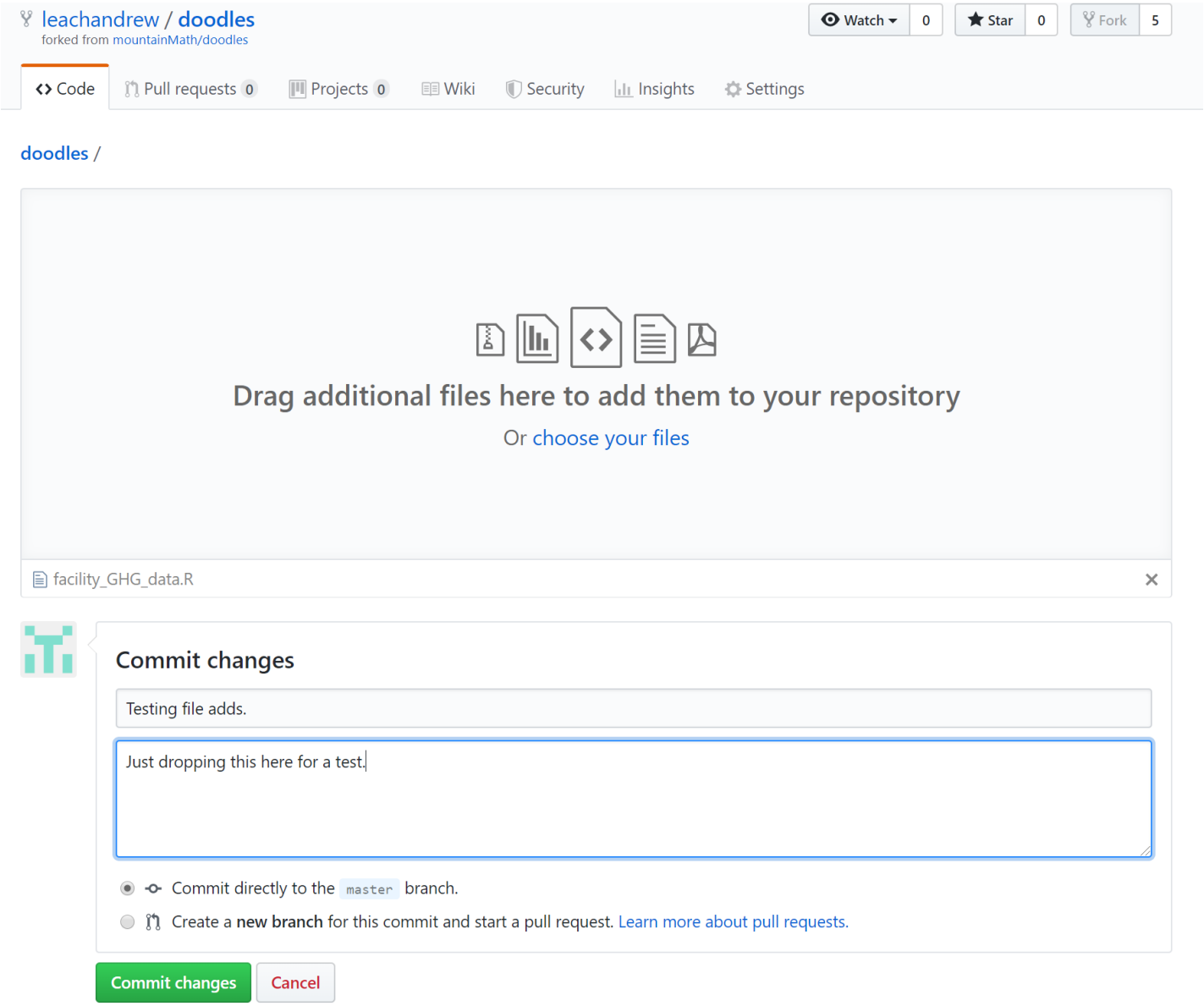


The repository contains all the files I've posted for you for use of Alberta power data. You want to then create a 'fork' to create a version of the files on your own GitHub repository. So, for example, if I want to use some files

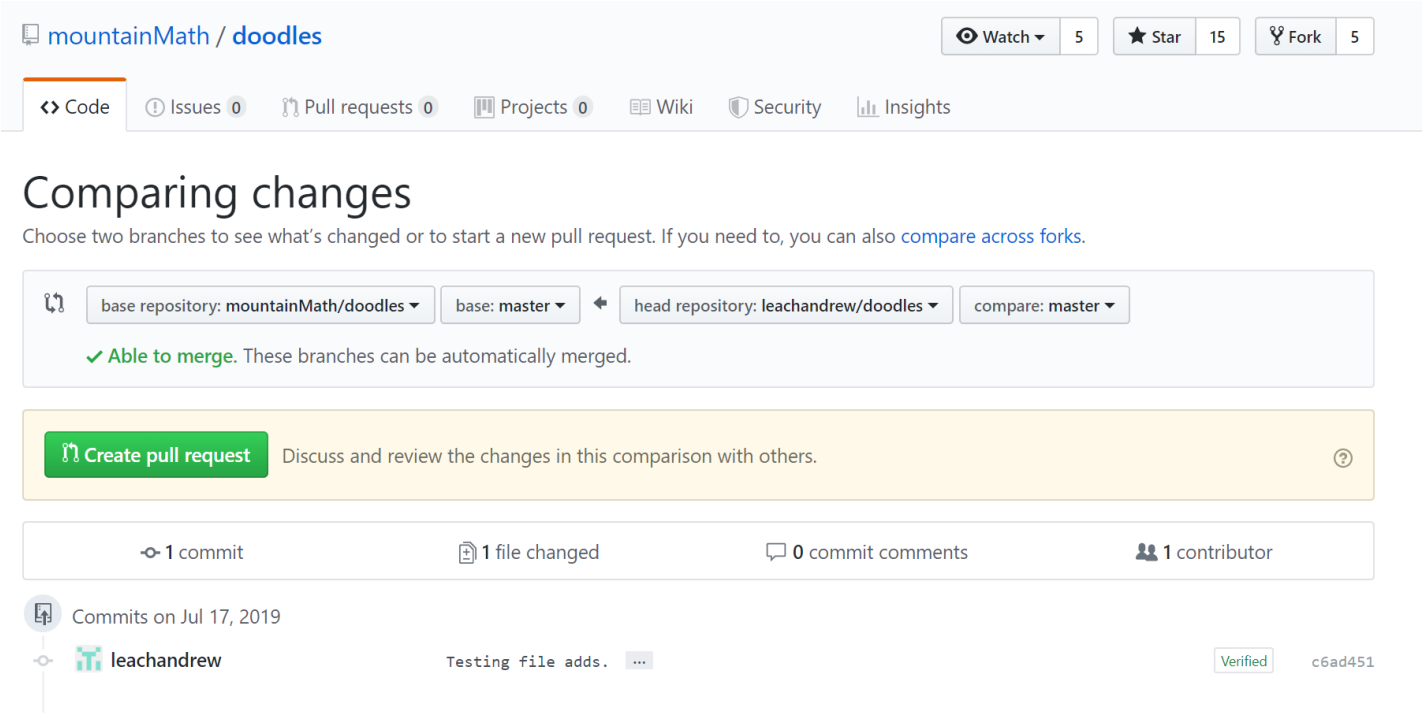
from Jens von Bergmann, I can go to his repository and create a fork of his Doodles repository:



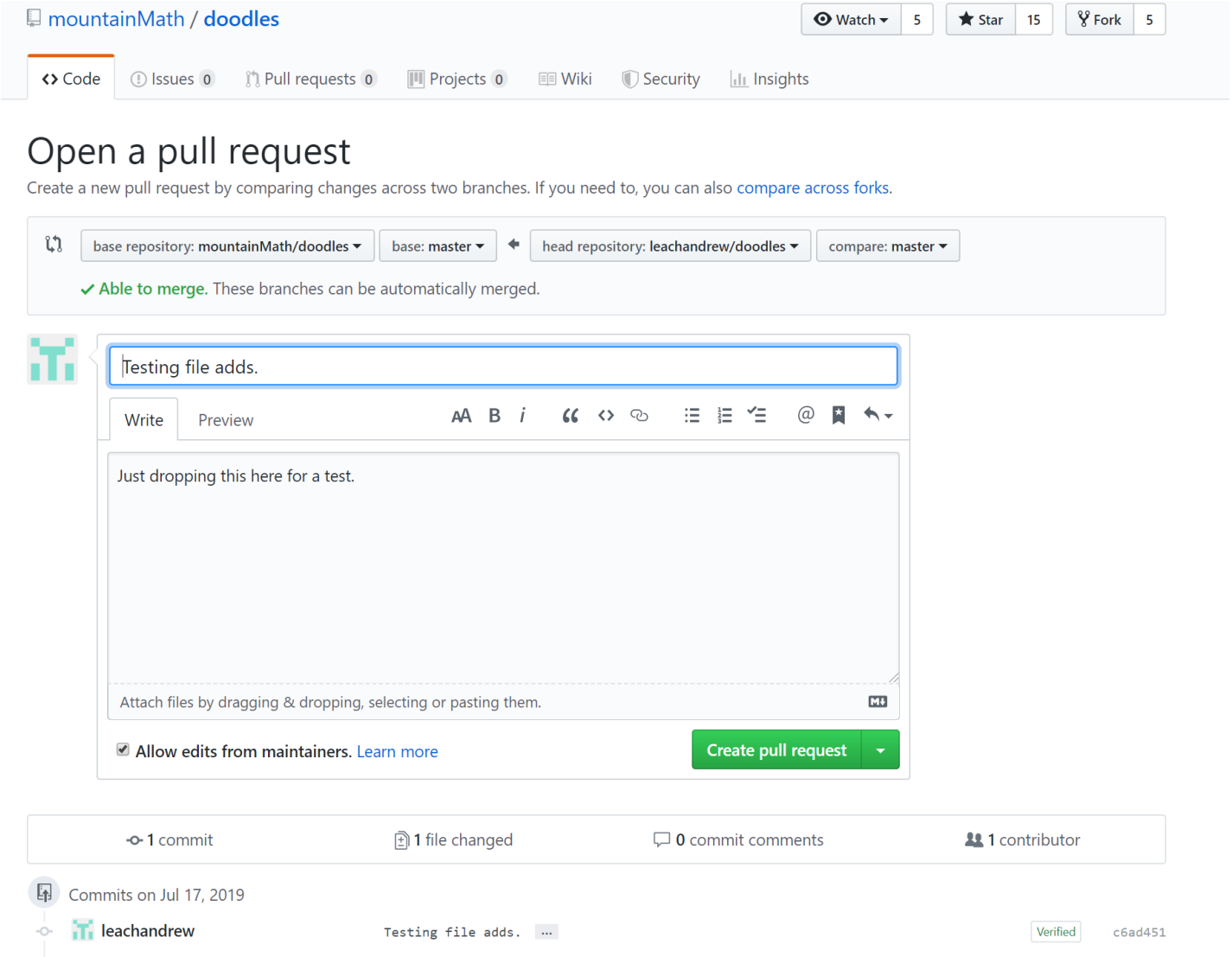
You can then download all the files in the repository using the Clone or Download link, depending on how you want to set things up on own home machine. Once you have the files on your machine, use them as you like, edit them, etc. but maintain the same file names. You can then, upload the files back to your forked repository. Name the commit (the upload) and make sure you add some comments about it to tell me what's going on.



When you've reached a point that you think the changes are useful for others, you can create a pull request. In this way, you're asking me to 'pull' your files into the master repository. In the case I've just built, it would look like this.

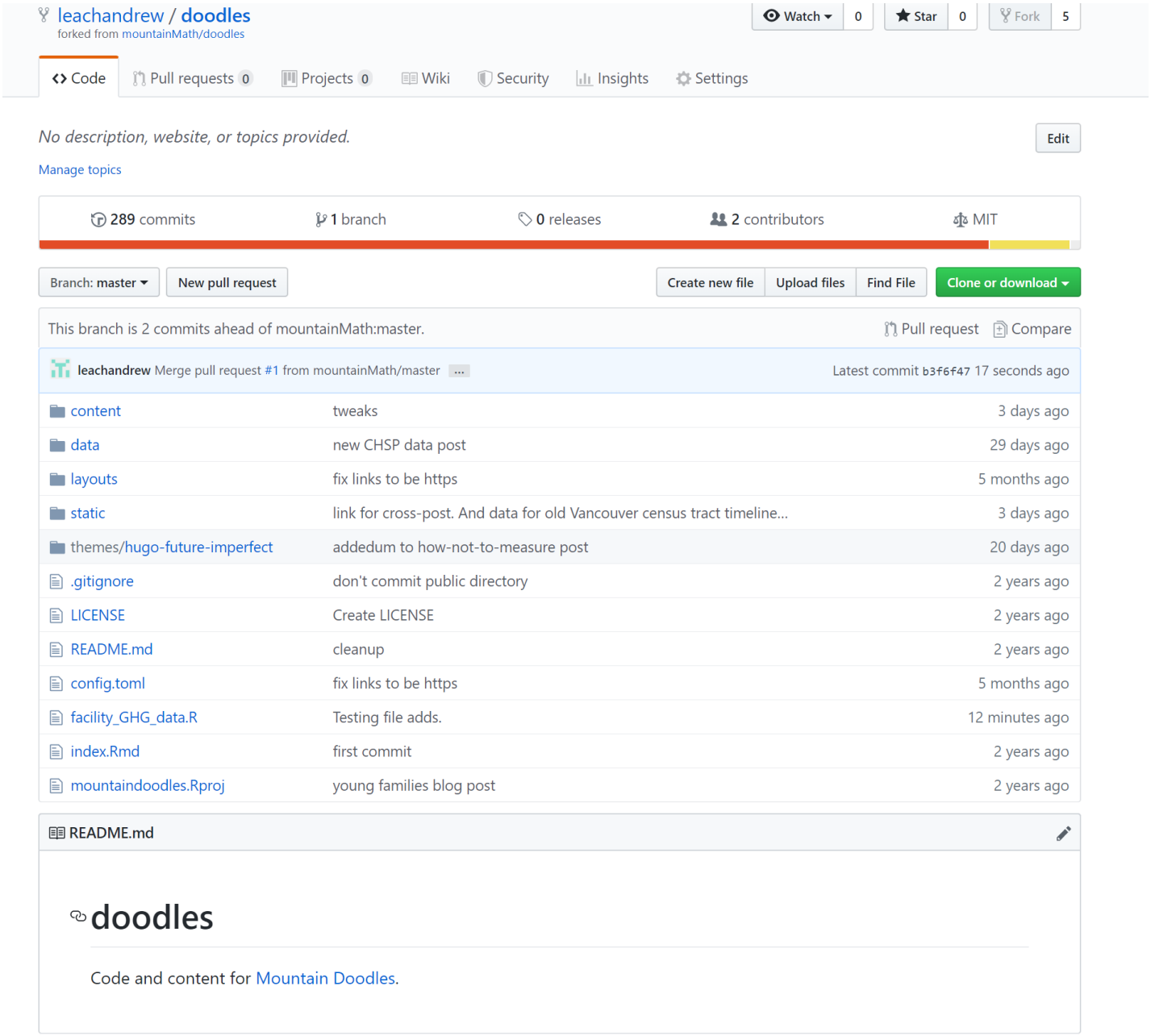


Once you've created the pull request, you'll get a screen like this where you can add more comments to explain what you've done. A good pull request would contain not only the changed files, but also some testing results that show what it does.



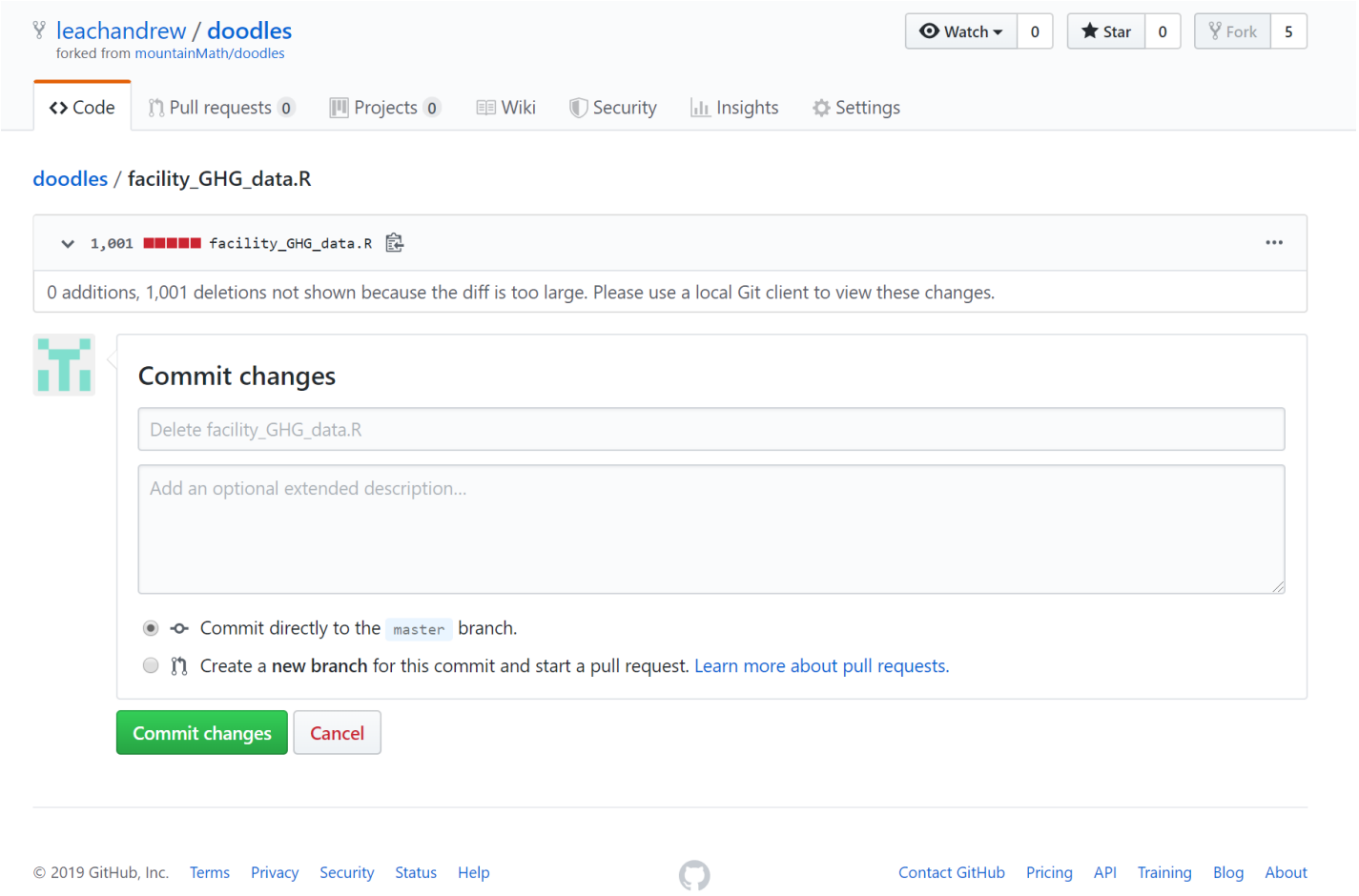
When you send me a pull request to assess the changes, I'll be able to see a line-by-line comparison, I'll be able to test the files, etc. If everything looks good, then I can merge your pull request into my master repository. The next time a member of the team downloads the repository, they'll get your modified version of the files.

This leads to an important point about repo management. If you look at my forked repo from Jens, it's way out-of-date, so I was at risk of modifying a file that's already been modified by Jens or others. AS a result, you want to make sure to keep your repo up-to-date. This is where GitHub is a bit cumbersome and not-at-all intuitive, but here are the instructions (https://github.com/KirstieJane/STEMMRoleModels/wiki/Syncing-your-fork-to-the-original-repository-via-the-browser) to do it via your browser only.



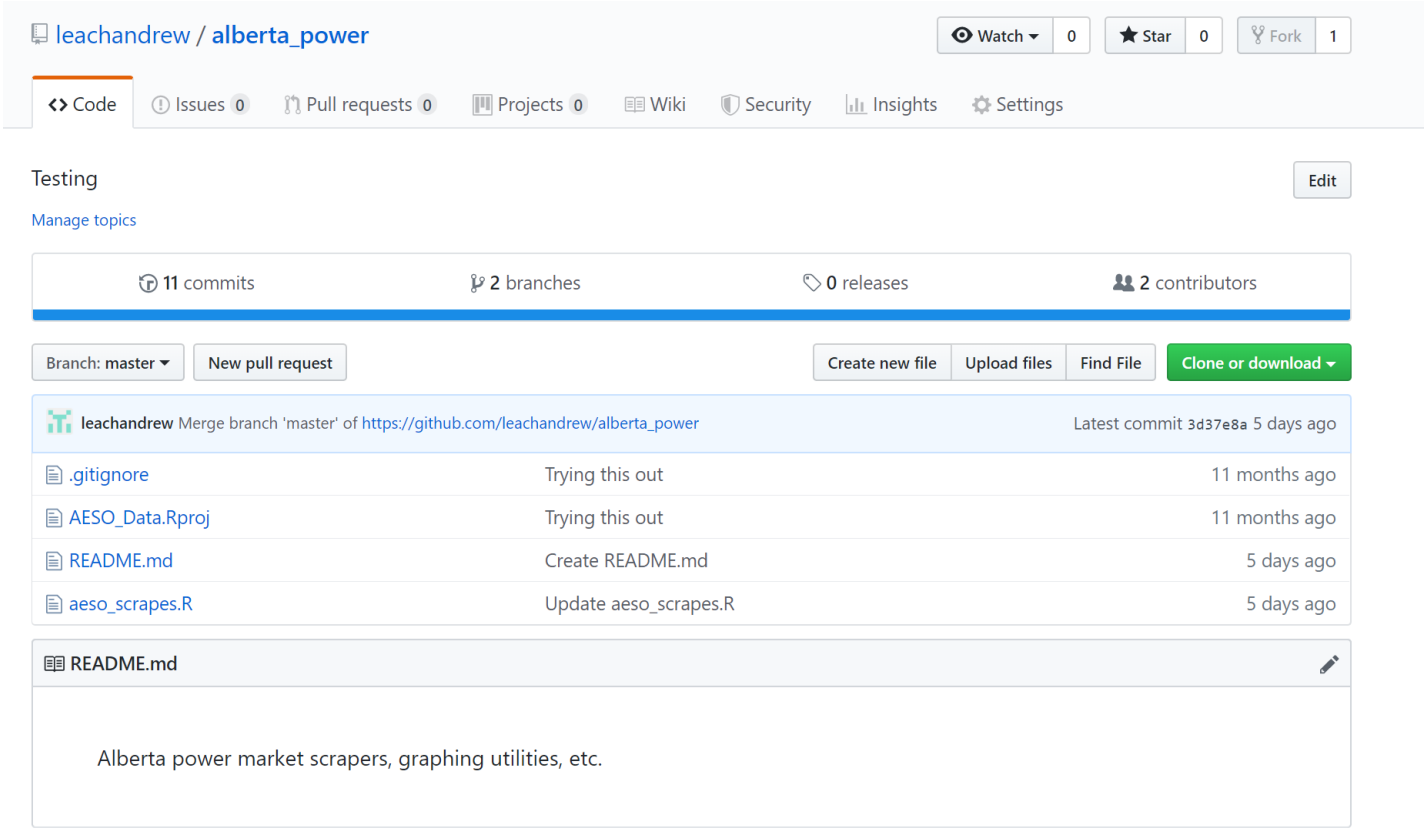
You can see that my fork is now up-to-date.

You can and should also use GitHub for version control. Unfortunately, you can't do that with the browser alone. So, for example, if I decided that I wanted to revert to a previous version of my online repository, I'd have to manually revert the file upload by clicking on the file and then clicking on the delete icon. Once you get there, you'll be asked if you want to 'commit' the changes, which you can do.



GitHub -> Your computer –> Send files back to me

This is the most basic workflow, and doesn't require much of a change from you. To access the files, log on to GitHub (https://github.com/leachandrew) and find the repository you want. For example, if you're looking for AESO scrapers, you could choose the alberta_power (https://github.com/leachandrew/alberta_power) repository.



The repository contains all the files I've posted for you for use of Alberta power data. You can download all the files in the repository using the Clone or Download link. Once you have the files on your machine, use them as you like, edit them, etc. but maintain the same file names. You can then, if you want to propose some changes, send me back the files you want to change and I'll deal with them on GitHub.