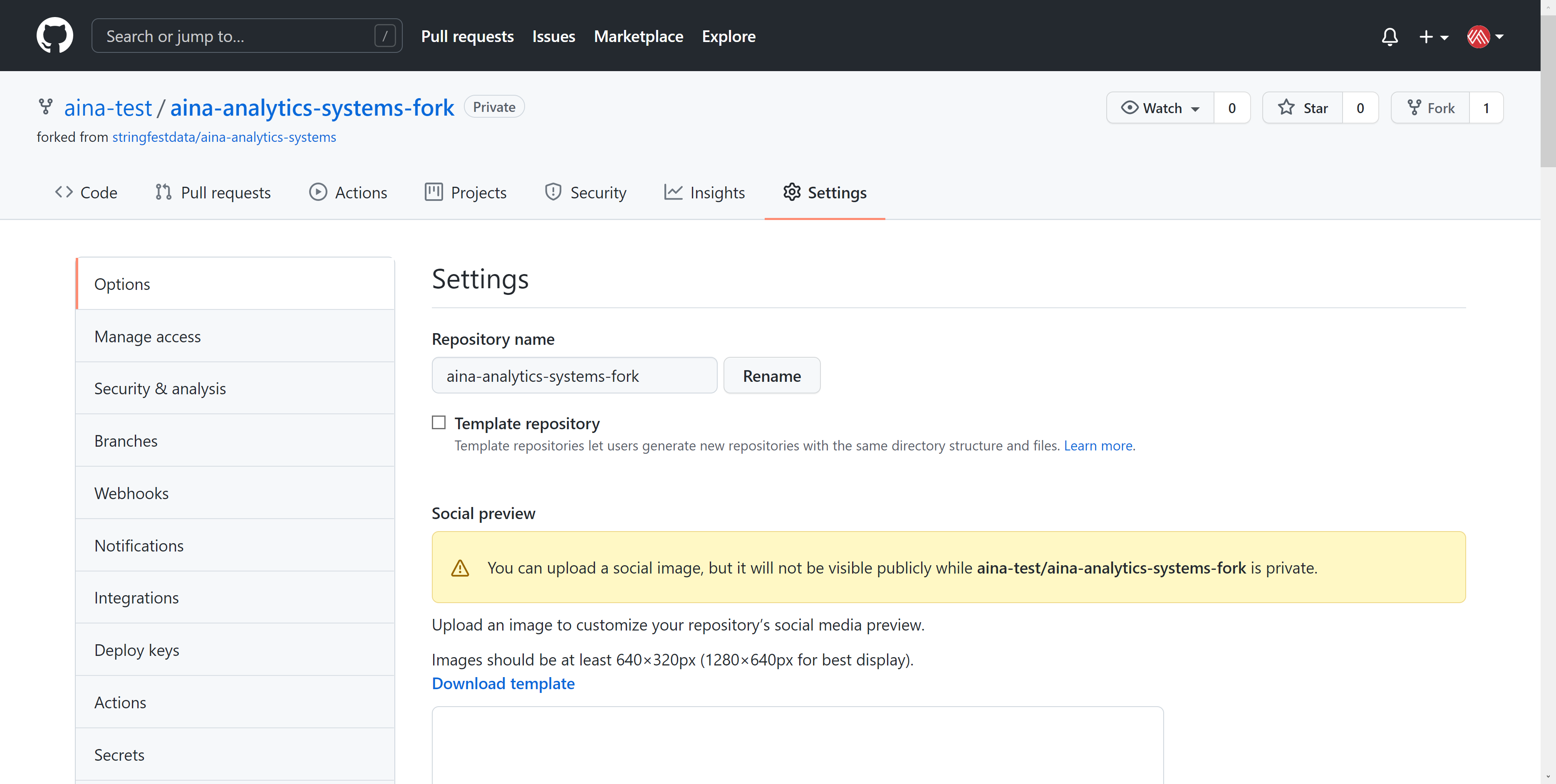
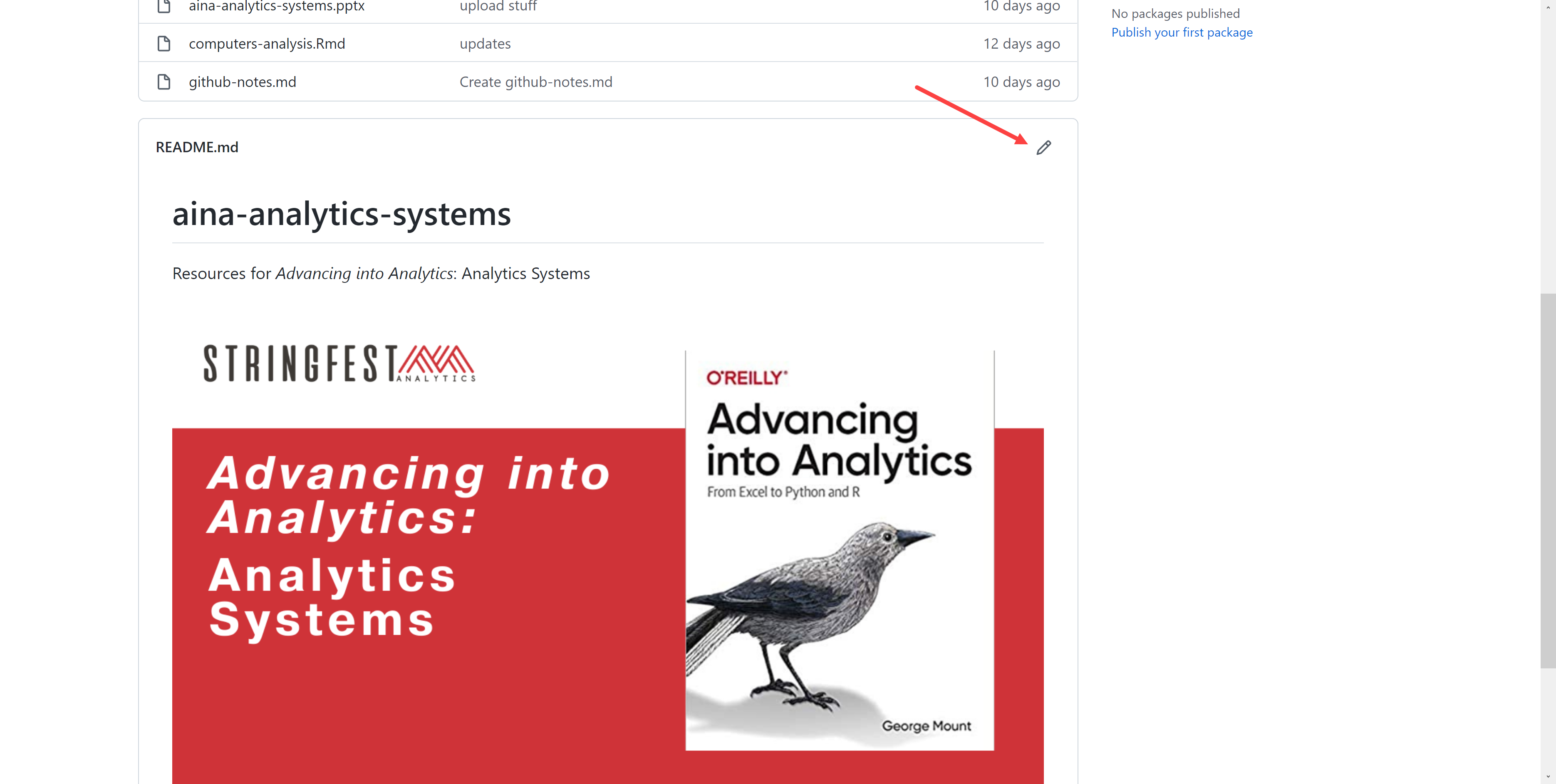
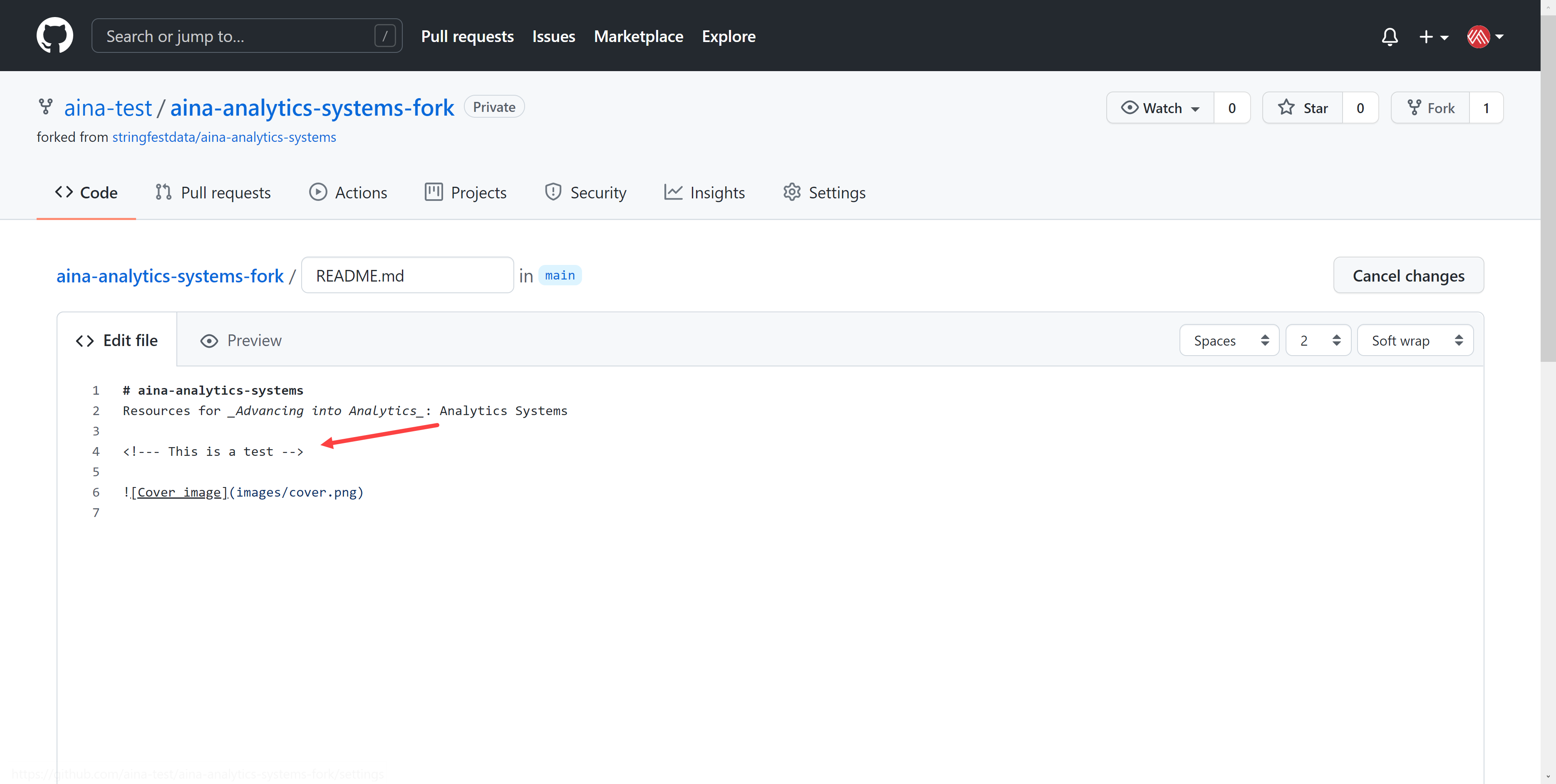
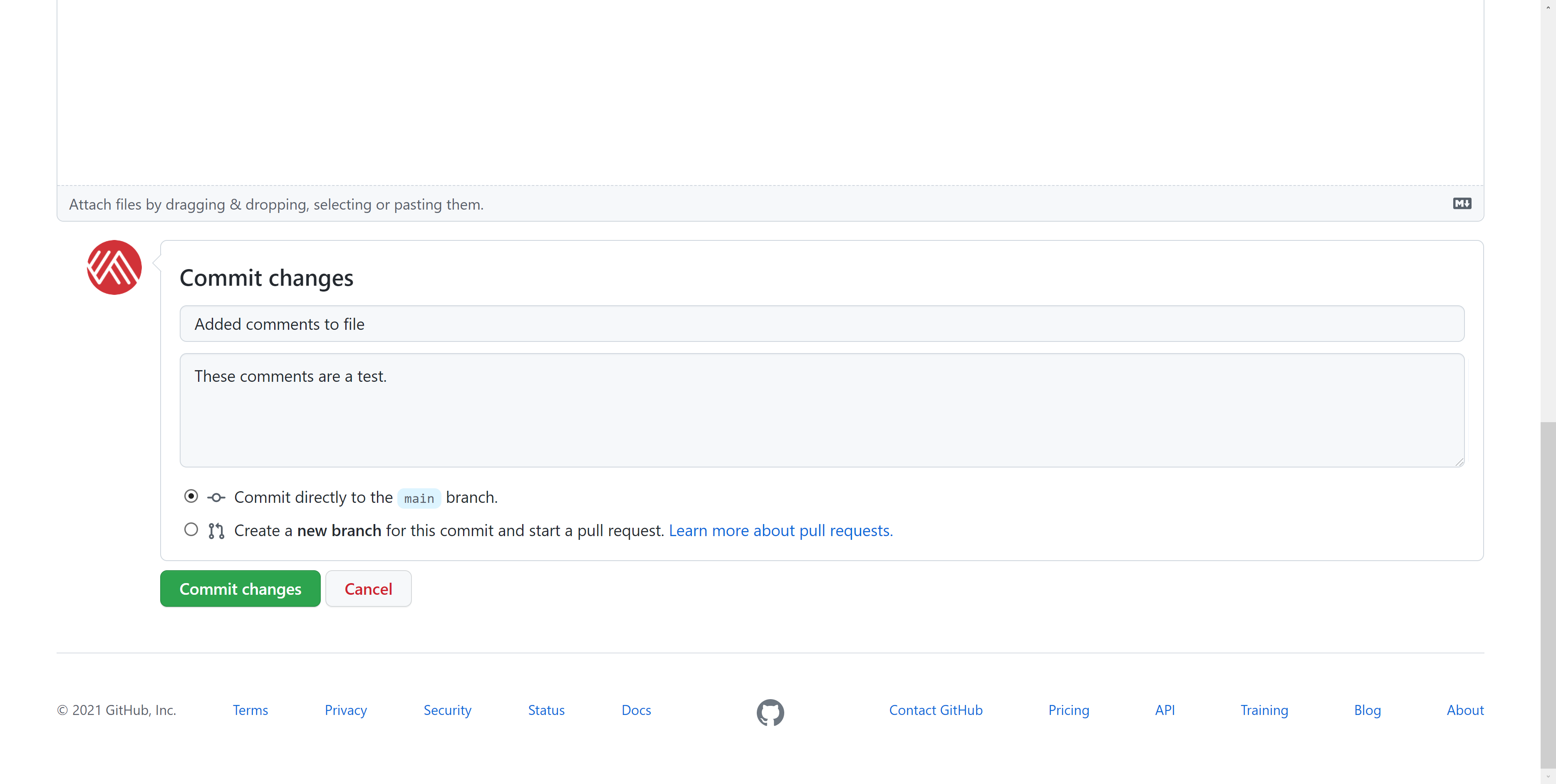
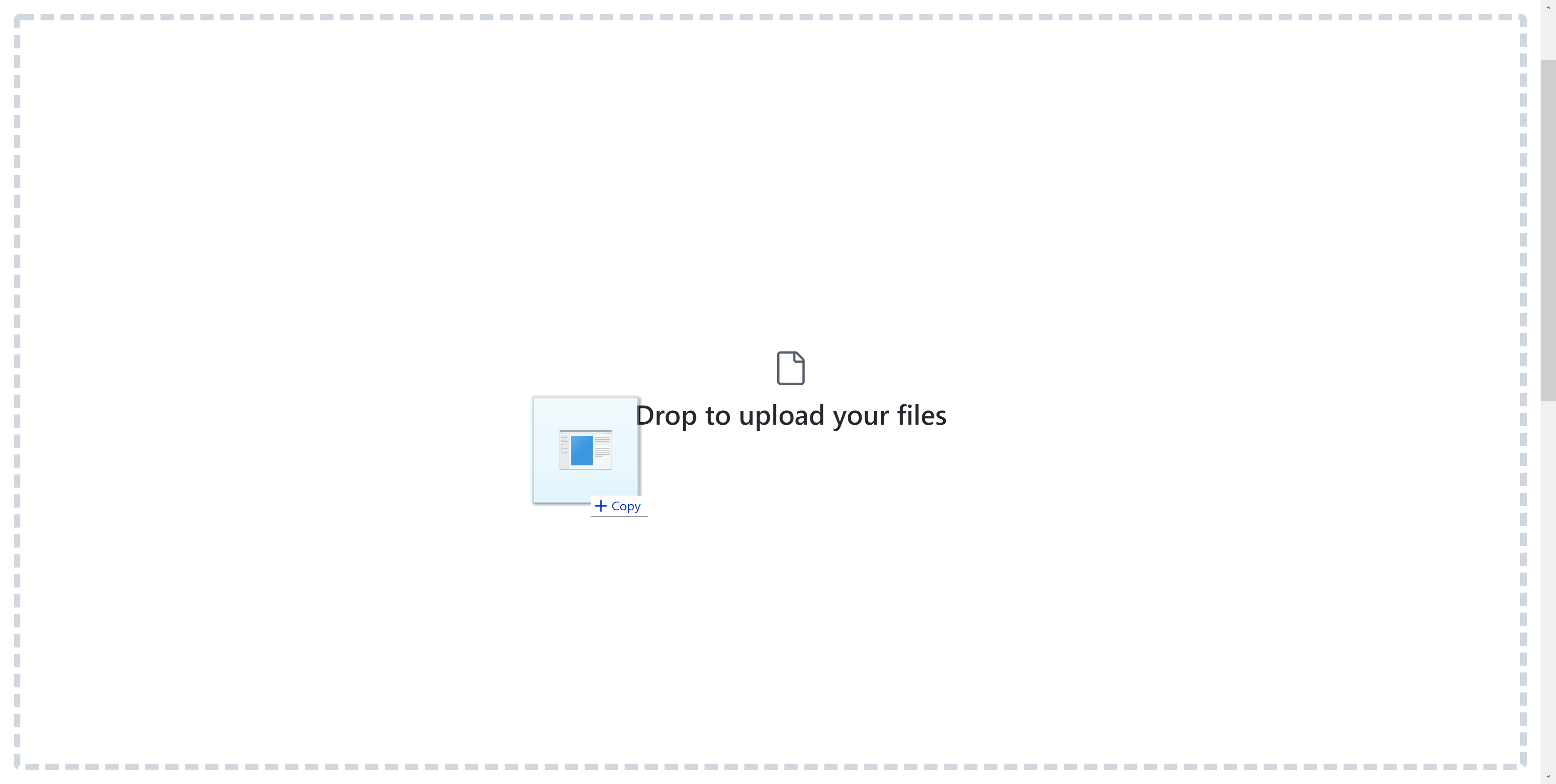
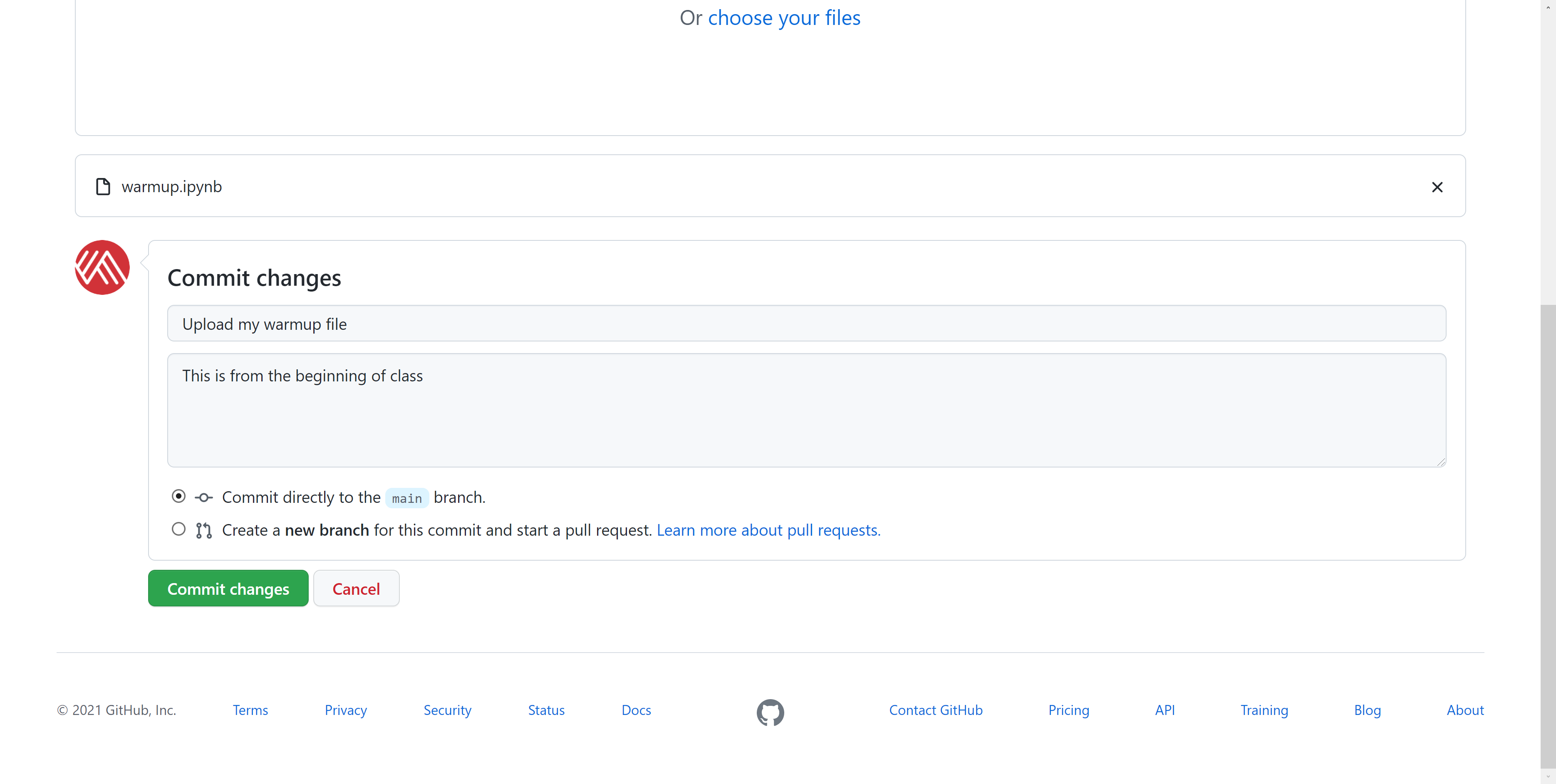
**GitHub.com basics**

1. I am going to rename my repo so it’s even more clear this is a forked repo. Go to Settings > Repository name:  
   

**Edit readme file**

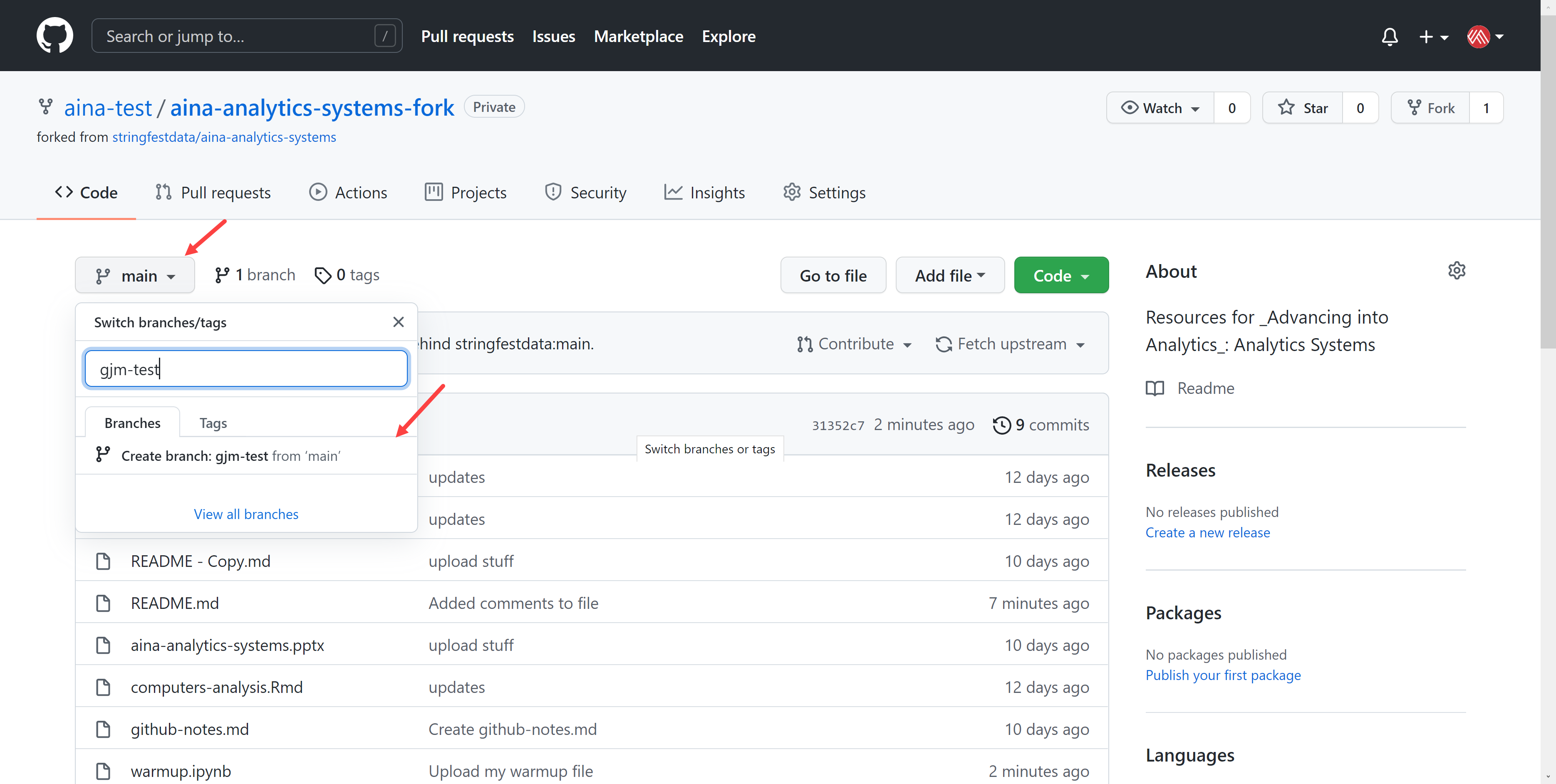
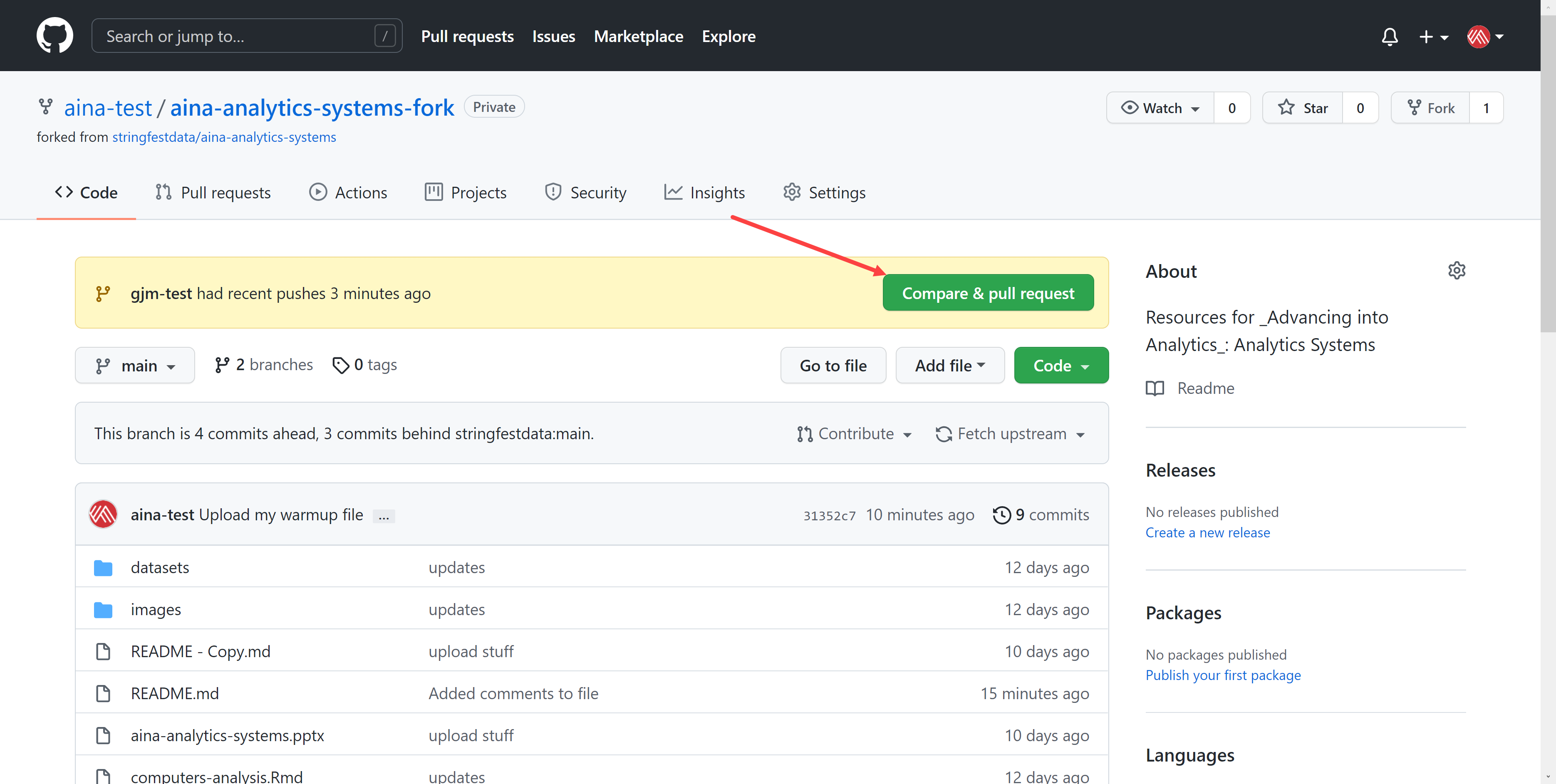
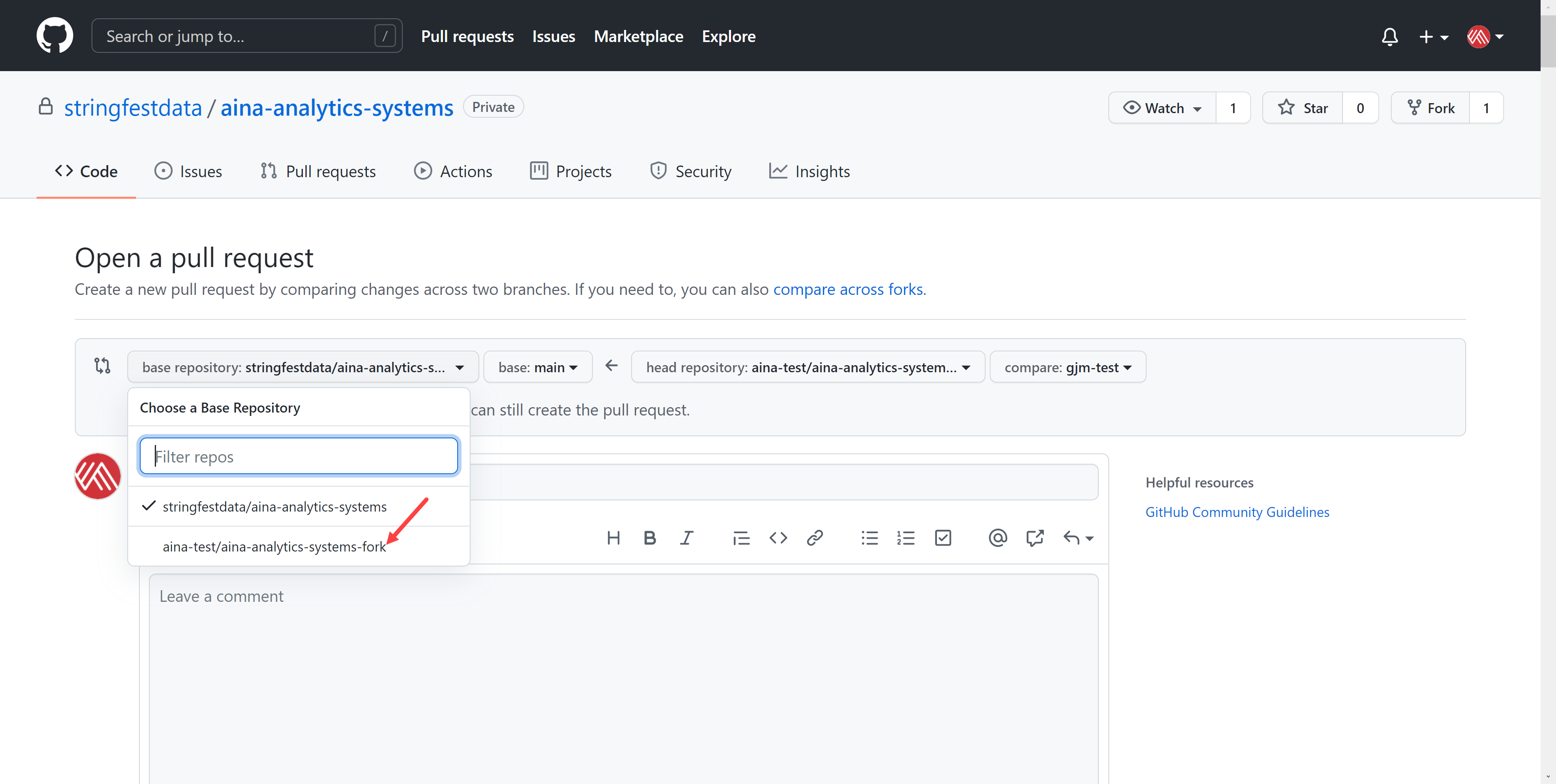
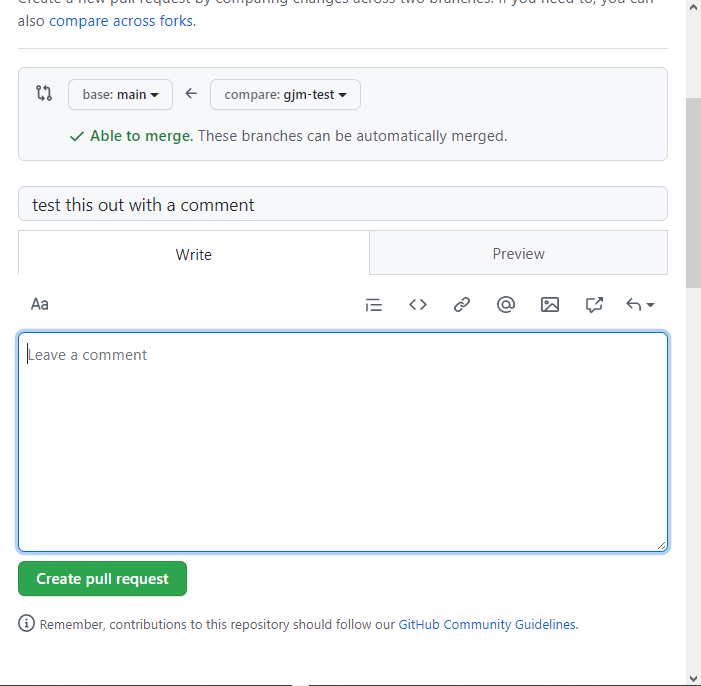
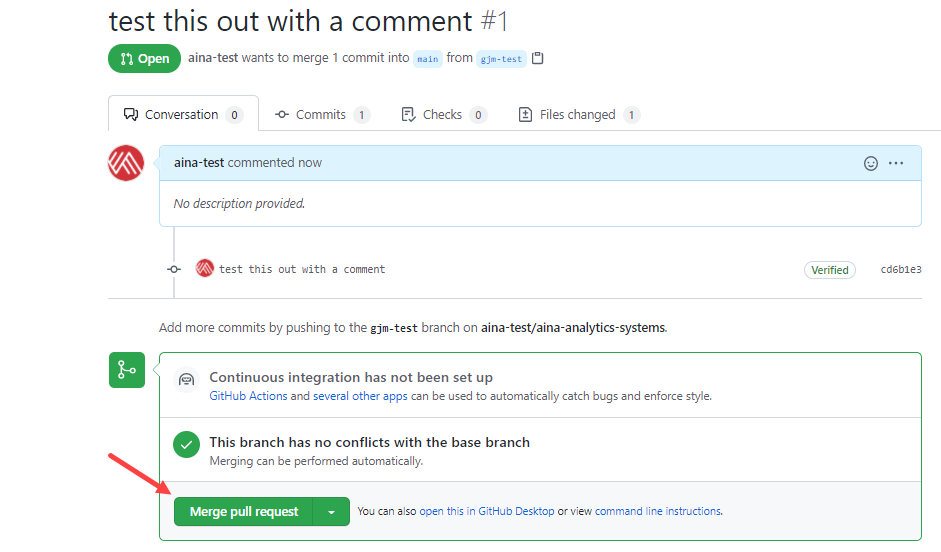
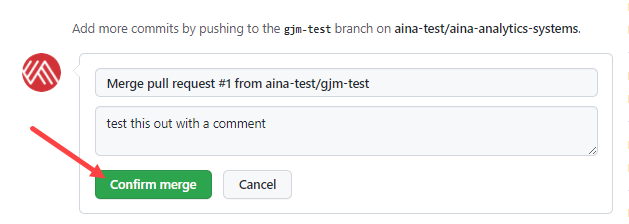
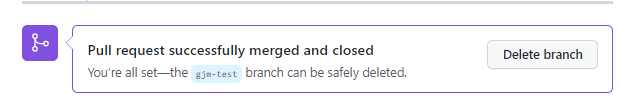
1. This README.md file is the Markdown file that by default renders as part of the repo. Click the pencil icon to edit it.   
   
2. Make some kind of change to the file. For example, you could just add some comments using HTML:  
   
3. Scroll to the bottom of the page. There will be a “Commit changes” menu. Best practice is to include a title and further explanation of what these commits are for, so they are more auditable.   
   
4. Click “Commit changes.” This change has now been committed to the readme file. You can edit other text- or code-based documents on GitHub using similar steps.

**Upload warmup.ipynb**

1. Drag and drop the file into the repo. Your screen should look like this:  
   
2. Add your commit message and commit the change:  
   
3. You can view Jupyter files as static pages in the GitHub repo, but you can’t run them and it’s *very* difficult to edit them. You’ll do much better off making those repo changes from your computer rather than on GitHub.com. We will discuss how to do that later on.

**Creating and merging pull requests**

While you could continue to commit directly to main, it’s a better practice to stage your changes in another branch, then merge your changes back in. This makes it easier for your team to review and audit changes to the main branch.

1. Go back to your repo’s homepage and click where it says main
   1. Give your new branch a name and click “Create branch.” You will now be on the new branch.   
      
2. Make some sort of change to the branch. For example, you could add some spaces or a comment to the readme file. Be sure to add a commit message!
3. Go back to your repo page. You should see a notification that gjm-test had recent pushes.
   1. Click “Compare & pull request”  
      
4. GitHub by default will try to merge these changes with the stringfestdata repo. Click the dropdown to choose your repo as the base to compare.   
   
5. You should see something like below. The original commit message from the change is included in the title. You can add more comments if you’d like.
   1. Click “Create pull request” when you’re ready.   
      
6. After that, the PR (#1!) will be opened. Click “Merge pull request” to merge these changes from gjm-test back into main.  
   
7. Finally, click “Confirm merge” to do this thing!   
   
8. You can generally delete the branch after this is done.   
   
9. Great work! 🙌 Return to the slides for additional GitHub knowledge.