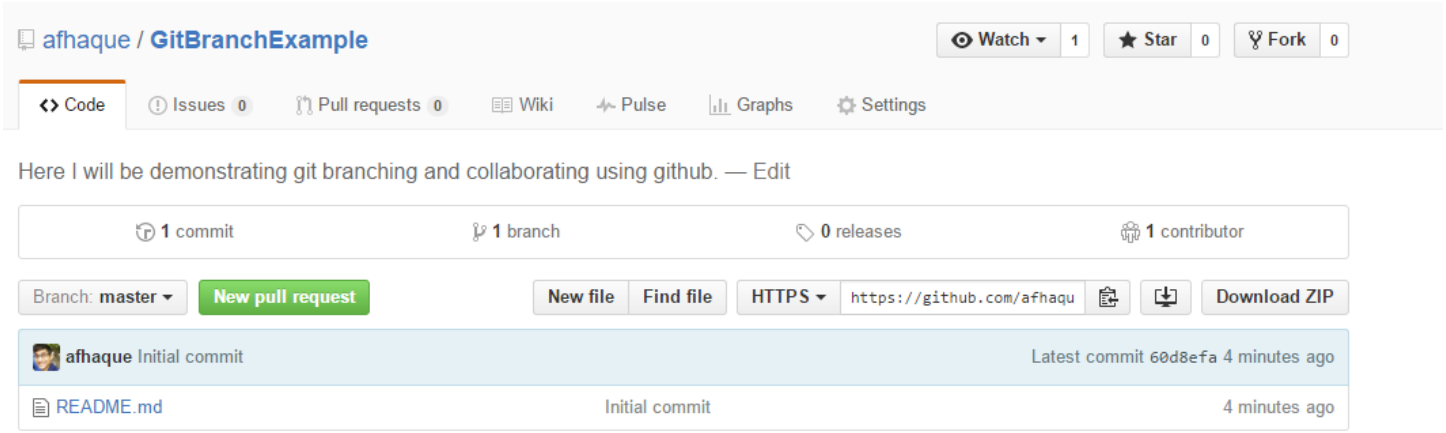
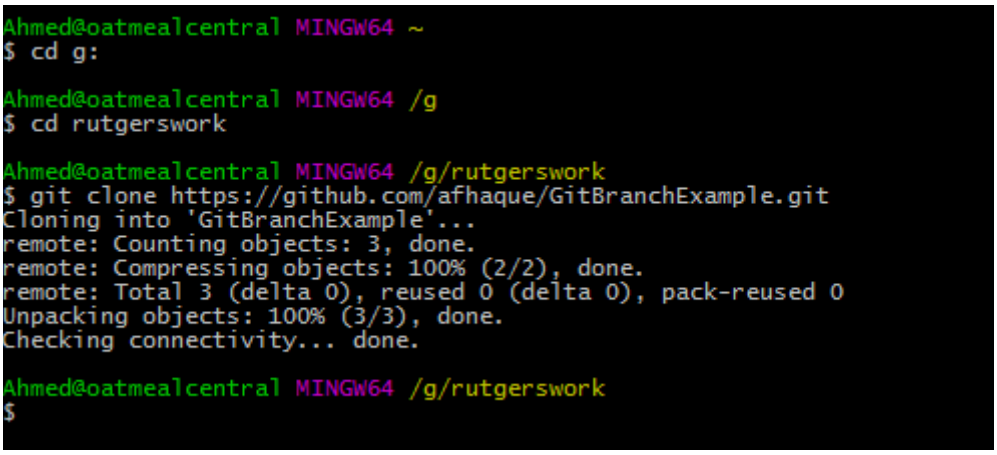
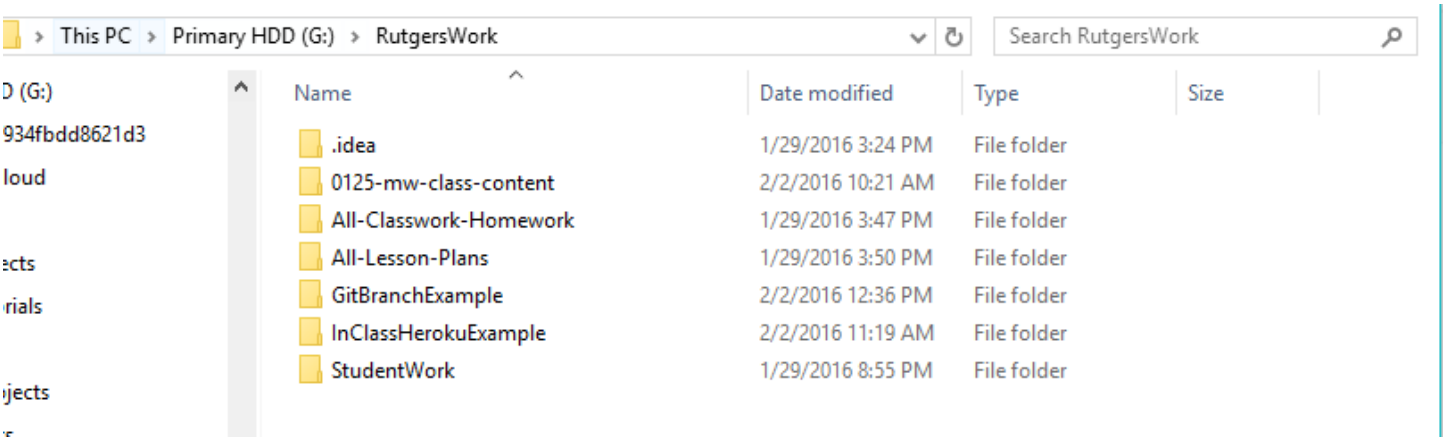
Create, Checkout, and Pull Git Merges

This guide will walk you through the process of creating branches on a single code repository. While this guide is intended for a single developer who would like to use pull requests on his/her own repository, it can be adapted to cases where multiple developers are working together. (See step for “Adding Collaborator”).

1. Find or create a code repository in GitHub. Clone this repository to your local directory if it hasn’t been cloned already. (In my case, I have a completely empty repository).







2. Navigate to this folder in git bash.

```
Ahmed@oatmealcentral MINGW64 /g/rutgerswork
$ cd GitBranchExample/

Ahmed@oatmealcentral MINGW64 /g/rutgerswork/GitBranchExample (master)
$
```

3. Run the following commands:
   git checkout master
   git pull

   In this example case, nothing will happen. It should simply say that you are already on master and that your branch is up-to-date. However, when collaborating this is an *incredibly important step*. Serious issues arise when two developers are working on "out-of-date" versions of code. If one developer's version is behind the other developer, it becomes very tricky to address code conflicts when merging. Don't let this happen if you can avoid it!!!!

```
Ahmed@oatmealcentral MINGW64 /g/rutgerswork/GitBranchExample (master)
$ git checkout master
Already on 'master'
Your branch is up-to-date with 'origin/master'.

Ahmed@oatmealcentral MINGW64 /g/rutgerswork/GitBranchExample (master)
$ git pull
Already up-to-date.
```

4. Next type git branch _____ (where you fill in the blank with a branch name of your choosing). Here we are telling git, "I'd like to make a new branch".

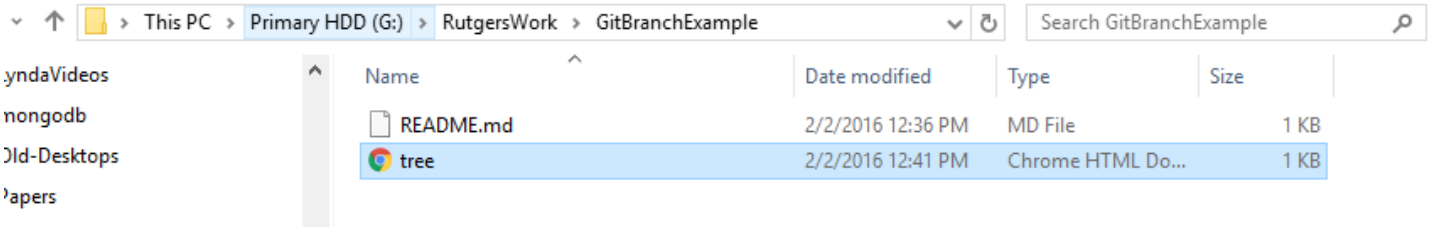
```
Ahmed@oatmealcentral MINGW64 /g/rutgerswork/GitBranchExample (master)
$ git branch ahmedsbranch
```

5. Next type git checkout _____ (fill in the blank with the name of the branch you just created). Here we are simply telling git, "I'm working in this branch now".

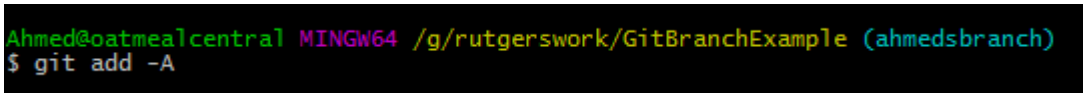
```
Ahmed@oatmealcentral MINGW64 /g/rutgerswork/GitBranchExample (master)
$ git checkout ahmedsbranch
Switched to branch 'ahmedsbranch'
```

6. Now, make some changes to the code inside the folder. In my case, I am going to add a file called tree.html that I made in Sublime.

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Trees are Awesome</title>
6  </head>
7  <body>
8      <h1>Trees are Awesome</h1>
9      <img src="https://upload.wikimedia.
       org/wikipedia/commons/e/eb/Ash_Tree_-_geograph.org.uk_-
       _590710.jpg" alt="">
10 </body>
11 </html>
```

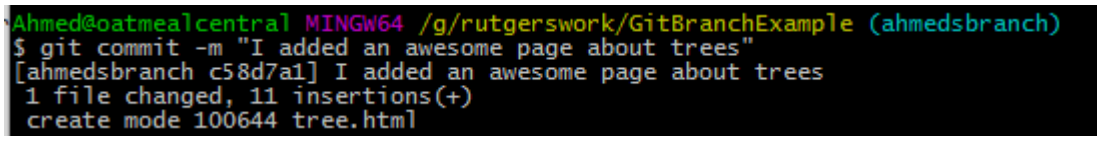


7. Next, go back to Git Bash and run the command git add –A. This is to tell git “Check for any and all changes I’ve made to the folder”.



```
Ahmed@oatmealcentral MINGW64 /g/rutgerswork/GitBranchExample (ahmedsbranch)
$ git add -A
```

8. Next, run the command git commit –m “_____” (where you fill in the space with a comment). This is to tell git “Confirm my changes locally”.

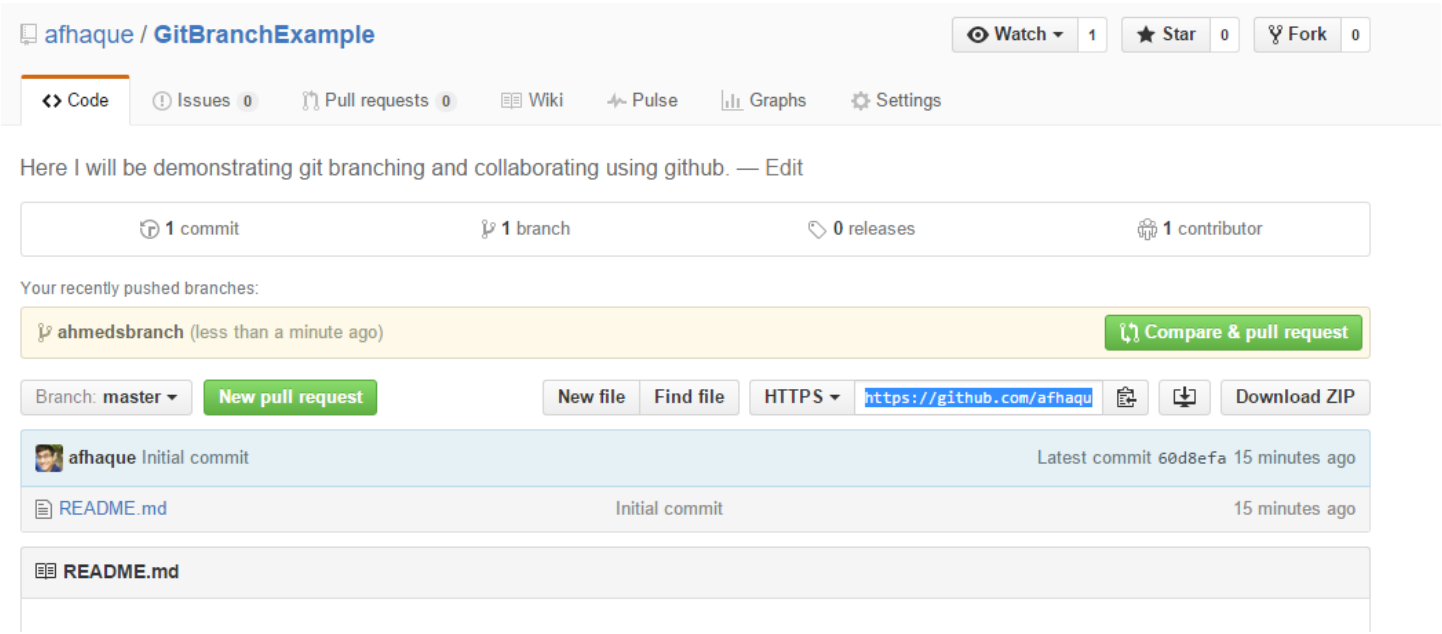


```
Ahmed@oatmealcentral MINGW64 /g/rutgerswork/GitBranchExample (ahmedsbranch)
$ git commit -m "I added an awesome page about trees"
[ahmedsbranch c58d7a1] I added an awesome page about trees
 1 file changed, 11 insertions(+)
 create mode 100644 tree.html
```

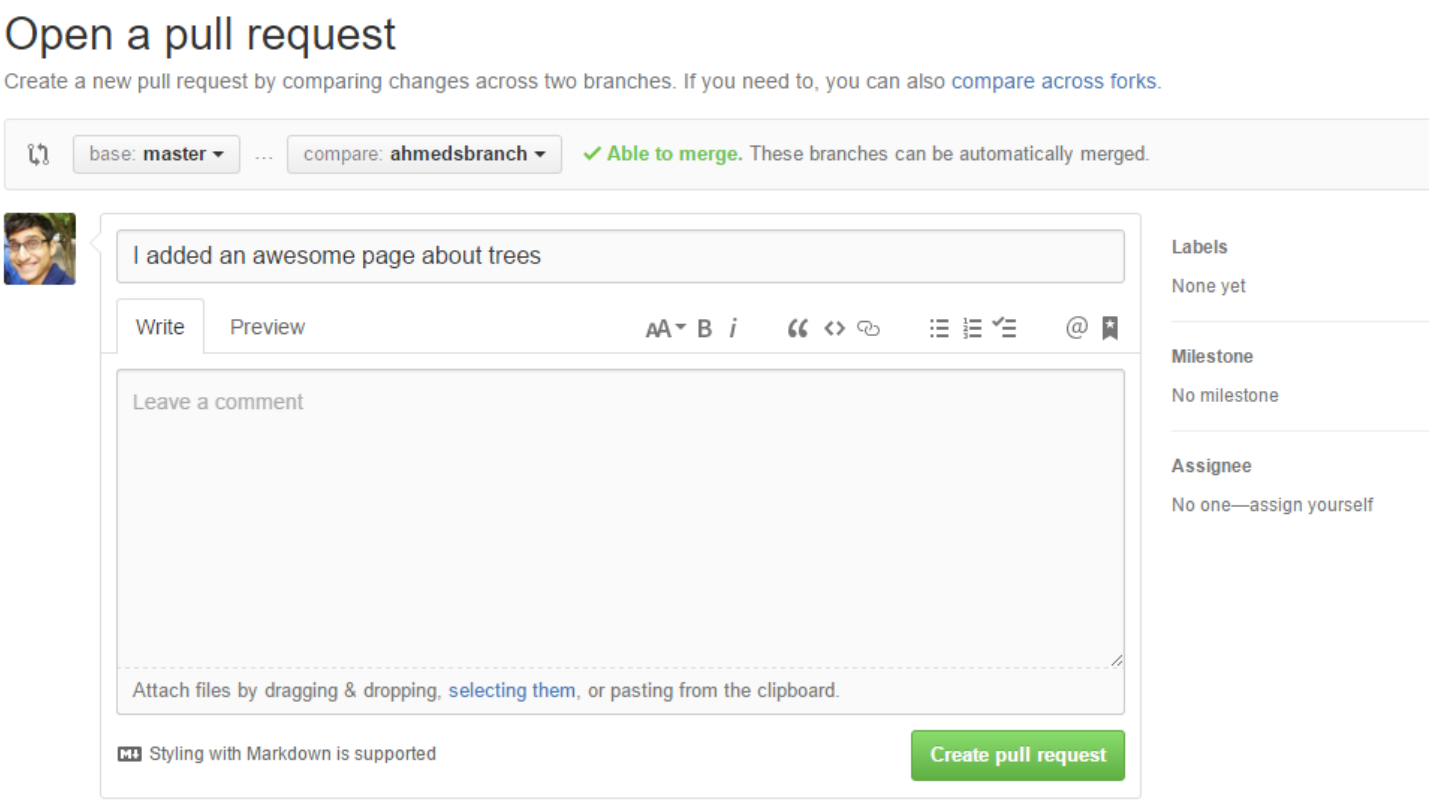
9. Next, run the command git push origin _____ (where you fill in the blank with the branch name from before). This will tell git “Upload all of my changes only to this specific branch”

```
Ahmed@oatmealcentral MINGW64 /g/rutgerswork/GitBranchExample (ahmedsbranch)
$ git push origin ahmedsbranch
Username for 'https://github.com': afhaque
Password for 'https://afhaque@github.com':
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 484 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/afhaque/GitBranchExample.git
 * [new branch]      ahmedsbranch -> ahmedsbranch
```

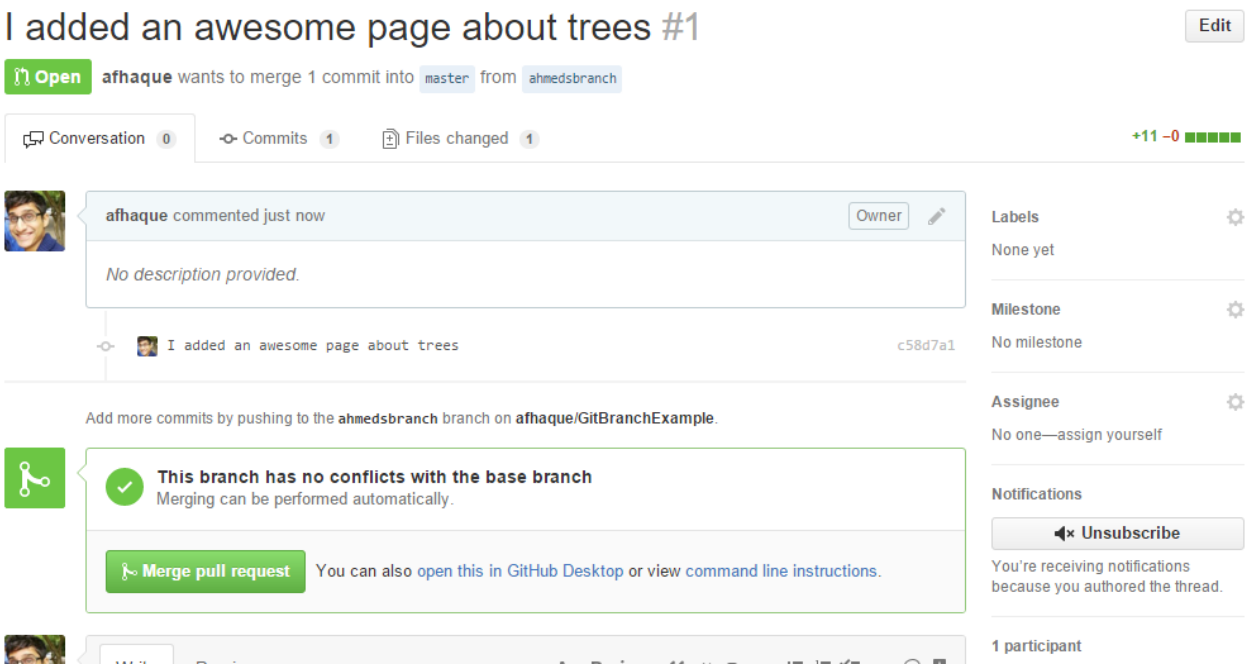
10. Now, navigate to Github.com and to the specific repository you are working with. You should see a new button with the label “Compare & Pull Request”). Click on it. This will take the information from the Branch and will request a “Pull Request”

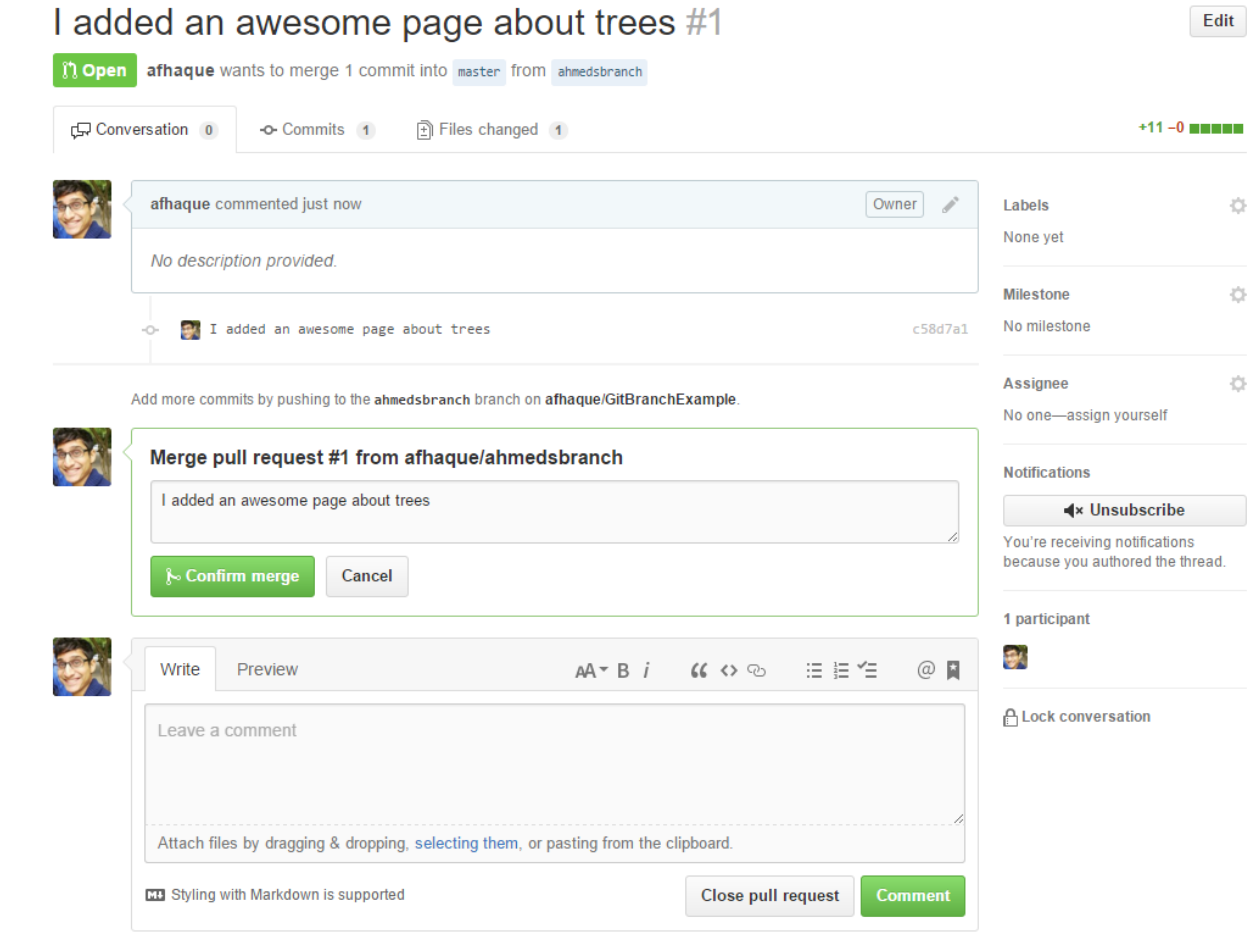


11. Hit “Create a Pull Request”. This will submit the requested update for the other person (in this case, still you) to accept.

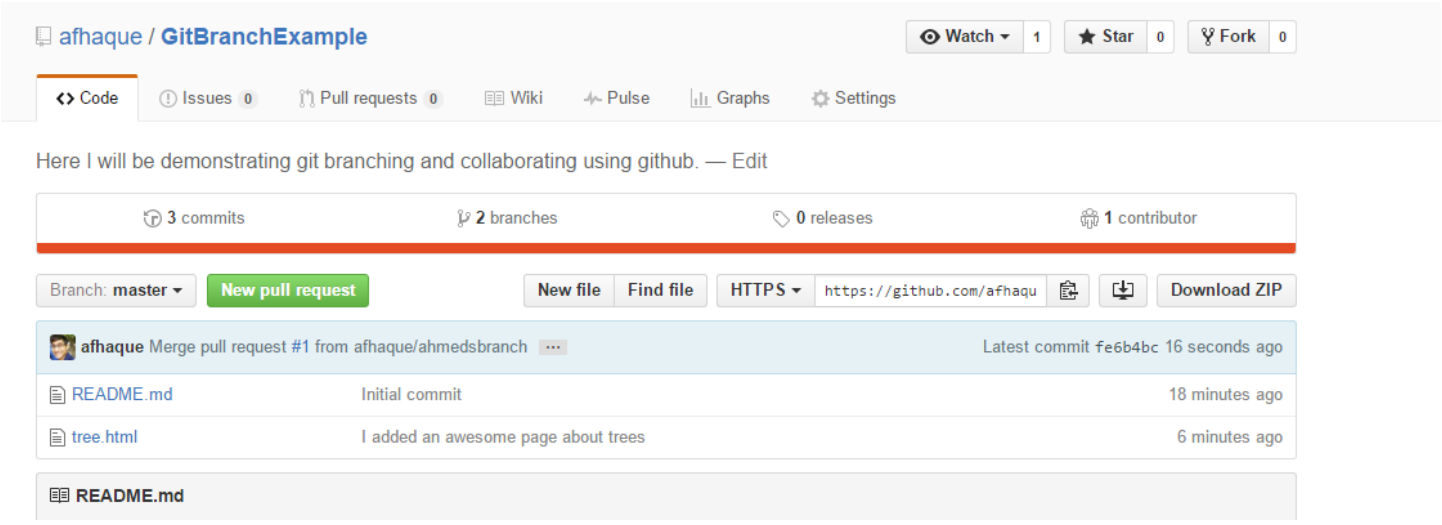


12. Next hit the button “merge pull request” and “Confirm Merge”.





13. Now if you go back into the repository you will see that the code has been incorporated.



VICTORY!!!

Collaborating Between Multiple Developers.

All of the steps mentioned above still apply for multiple developers. The only differences is that:

1. Developer #1 must copy the Git URL of Developer #2.

2. Developer #1 must add Developer #2 as a collaborator in Github. To do this go to Github.com. Navigate to the repository you'd like to collaborate on and go to settings. Then add Developer # as a collaborator.