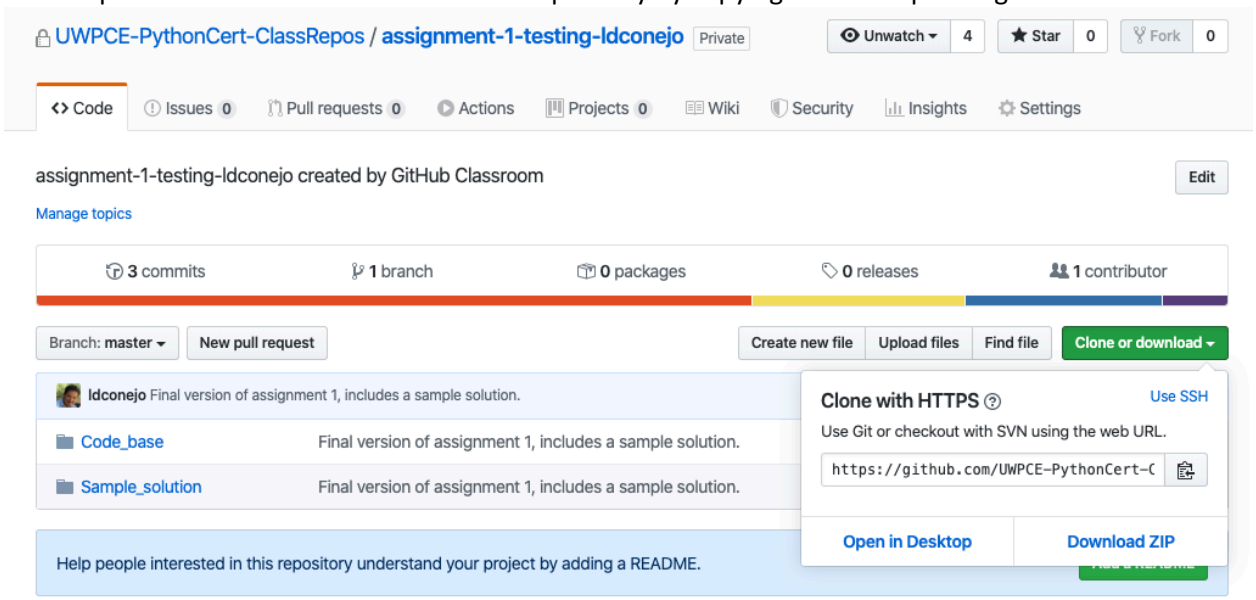
Working, submitting & reviewing assignments

1. This guide assumes a student has already accepted the invitation to work on an assignment, triggering the creation of an individual, private repository.
2. First step for the student is to checkout the repository by copying the corresponding link.



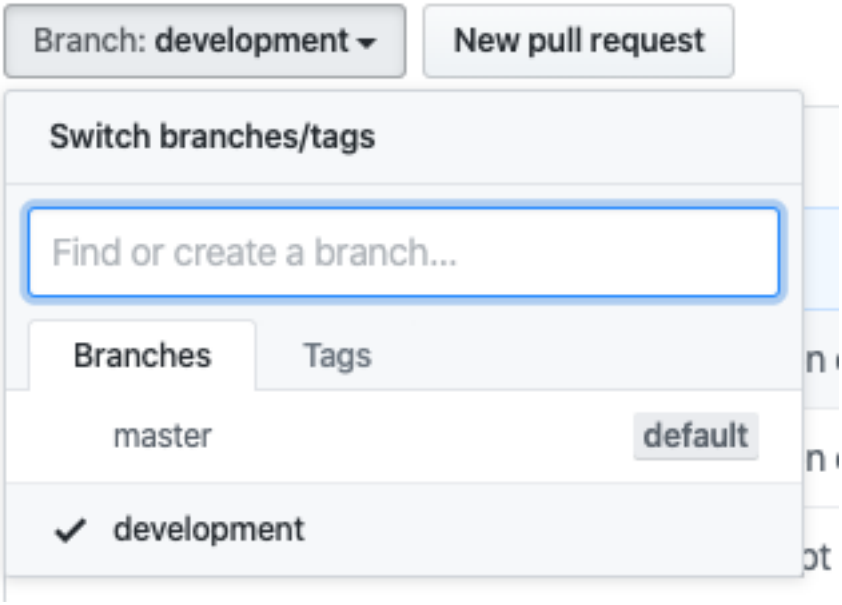
3. Clone the repository in the local folder selected for this purpose in the student's computer. For this example, the command would be:
   `git clone https://github.com/UWPCE-PythonCert-ClassRepos/assignment-1-testing-ldconejo.git`

4. This will create a new folder with the repository. In this case, the name of the folder is *assignment-1-testing-ldconejo*.
5. It is recommended that **the students create a branch and use it to work on their code**. Doing this will also make it possible for them to **create a pull request against their own private repository once their assignment is ready for submission**. Otherwise, if they work directly on *master*, no pull request can be created (the instructor would still be able to see the code, but some features like code review would not be available).
6. Create a new branch called *development*:
   `git branch development`
7. Switch to the new branch:
   `git checkout development`
8. Typing *git branch* should show *development* as the active branch:

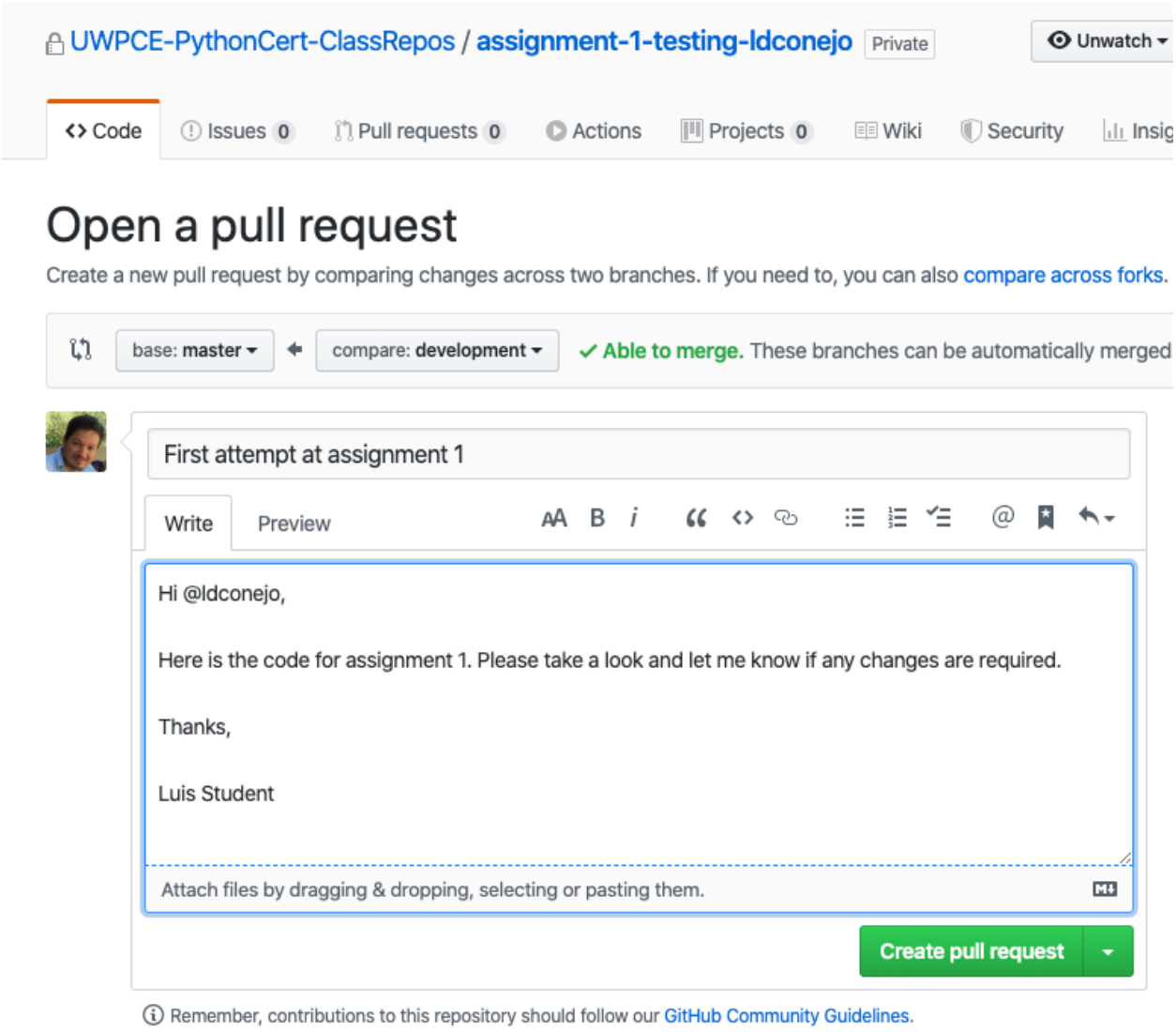
```
ldconejo-mac09:assignment-1-testing-ldconejo ldconejo$ git branch
* development
  master
```

9. In this example, we have added a new folder, called *submission*, and populated it with several files, representing a student's work on the assignment.
10. Inside *submission*, run *git add .* to include all files in the folder. Then run *git commit* to create a new commit. Enter a descriptive name for the code being committed.

11. The first time you push, you will need to tell Github the upstream repository to be used (note *development* is the name of the branch, if a different name is used the command must be updated accordingly):
    `git push --set-upstream origin development`
12. After this initial *push*, you will only need to run *git push* to commit your changes to the remote repository.
13. In the student's repository page, the new branch is now visible.



14. When the student is ready to submit their code for review or when they want to get feedback from the instructor, they can create a new pull request from the *development* branch into the *master* branch:



15. Click on *Create pull request*. At this point, all admins for the classroom (usually, the instructor and the TA) will get a notification that there is new pull request waiting for review. The rest of the process is the same as before, with Github allowing you to comment on the code, start a formal review, request changes, reject a pull request, etc.
16. Key advantages:
    a. The repositories are private, only admins for *UWPCE-PythonCert-ClassRepos* can see all of them. Students can only see their own repositories.
    b. The original assignment repositories cannot be accidentally modified by the students (Exception: If the original repository is created as public it would technically be possible for *any* Github user to make changes and submit a pull request to have those changes incorporated, but one of the admins to *UWPCE-PythonCert-ClassRepos* would need to approve said pull request).
    c. We retain all Github features for code review.
    d. As a future convenience, this could be linked with Canvas to automatically reflect grading of an assignment.

ldconejo (3/11/2020)