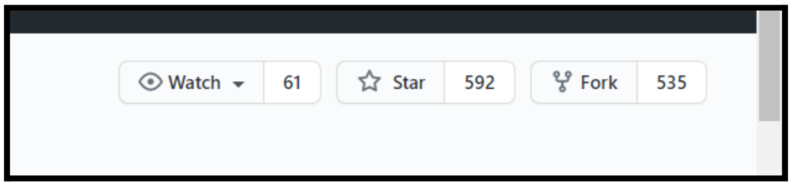
Quick Intro to git

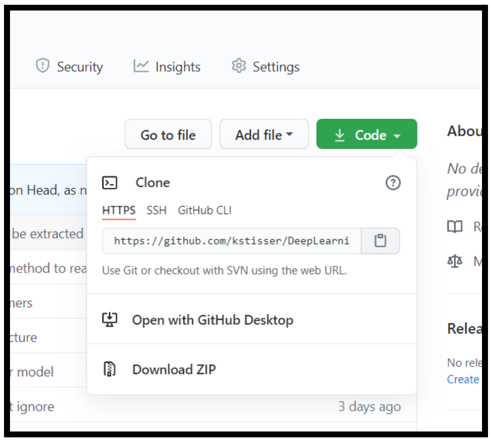
Step 1: Fork & Clone the repo

- Ensure git bash is installed if you are on a Windows machine
- If you are on a Mac or Linux machine you should already have it installed
- First in (linux/Mac) a terminal cd to the directory you'd like to clone your repo to (ie cd Documents). If you are on Windows right click the file location you want your repo saved to and click 'Git Bash here', and a terminal will open.
- Note- all commands referenced will be typed in the above said terminal when interacting with git

In github click the Fork button in the upper right corner of the repo:
https://github.com/kstisser/DeepLearningPointCloud



This should create a copy of the project in your repo
Command 1- Clone the repo (replace with your github username):
git clone https://github.com/<your github username>/DeepLearningPointCloud.git
Note- you can copy the https… value from your repo if you click on the green code button:



https://github.com/kstisser/DeepLearni

Use Git or checkout with SVN using the web URL.

Once you type the above command it should clone the repo. It may ask you for your credentials to github to be able to proceed. Now, to be able to run any git commands you need to cd into the repo, so type:
cd DeepLearningPointCloud

Now, we are going to add the original project as a remote:
git remote add upstream https://github.com/kstisser/DeepLearningPointCloud

You should be able to see this worked by typing:
git remote -v

Step 2: Create a branch
Important: NEVER COMMIT ANYTHING TO THE ‘main’ BRANCH, ALWAYS CHECKOUT YOUR OWN BRANCH
To create your own git branch (this is a way you can code on your own in the repo without affecting anyone else until you are ready to merge your code in), type the following:

```
git checkout -b <branchName>
```

For example:

```
git checkout -b clusterProfiling
```

Your branch name should have to do with whatever you are developing. There should be no spaces in your branch name.

Step 3: Code (this is your own step)

Step 4: Ok, so you’re ready to make a commit because your code works! Commit and push up. First, add each file to the staging area.

```
git diff <filename>
```

This will tell you the differences in the file you changed. If the file is red and not yet staged and **you don’t want those differences**, to get rid of them, type:

```
git checkout <filename>
```

If the file is green and staged and **you don’t want the file to be staged**, to put it back in the working directory, type:

```
git restore --staged <filename>
```

If you do want to add your changes:
To stage them in your working directory:

```
git add <filename>
```

Once you have all files you want staged, commit them all:

```
git commit -m “This should be a short comment about what change you made in your commit”
```

Now, you want to make it available to others remotely to be able to pull your changes, so push it to github:

```
git push origin <yourBranchName>
```

Step 5: To make sure you are up to date with what is on the main branch (you should do this when you are in a good working state- you have committed things):

```
git pull upstream main
```

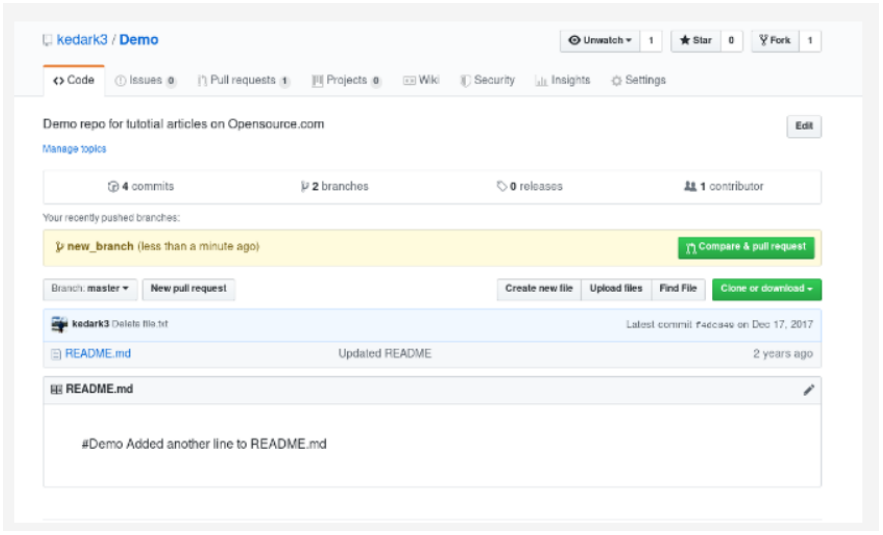
If it complains about a merge conflict, feel free to call for help and I’ll walk you through it. If this happens and you don’t want to wait, for now do:

```
git merge --abort
```

Step 6: Just a note of another useful tool to see what your status is:

```
git status
```

Step 7: When you are ready to merge your branch in:
Click on the Compare and Pull request in your github project:



Demo repo for tutotial articles on Opensource.com

Your recently pushed branches:

new_branch (less than a minute ago)

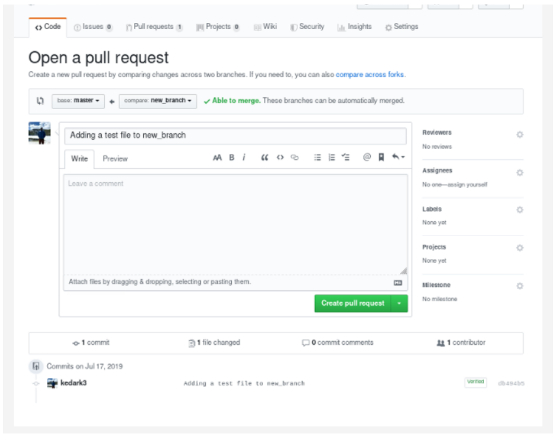
kedark3 Delete file.txt — Latest commit f4dc84e on Dec 17, 2017

README.md — Updated README — 2 years ago

README.md

#Demo Added another line to README.md

Fill in the following page. You should be looking to merge into the main branch. Write a description on what changes you are making. I will review the changes once you submit. **Make sure you have done step 5 (pulling main) and have tested before doing this.**



Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also compare across forks.

✓ Able to merge. These branches can be automatically merged.

Adding a test file to new_branch

Attach files by dragging & dropping, selecting or pasting them.

Commits on Jul 17, 2019

kedark3 Adding a test file to new_branch

There are many other git commands, please continue to explore! Just be careful with git reset.

Here is a simple reference for some of these steps:
https://opensource.com/article/19/7/create-pull-request-github