Using Pull Requests with Forks

Forking creates a copy of a repository which you can work on. You can then submit a pull request to have your changes reviewed and integrated with the master repo. This is the order of operations:
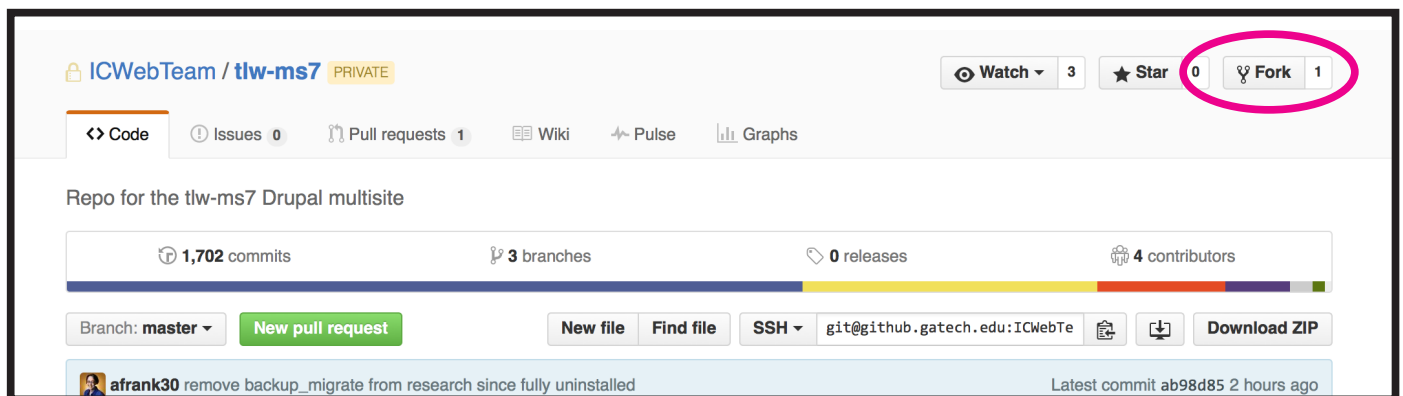
FORK > BRANCH > COMMIT > PUSH > PULL REQUEST

FORK

1. Fork the repository
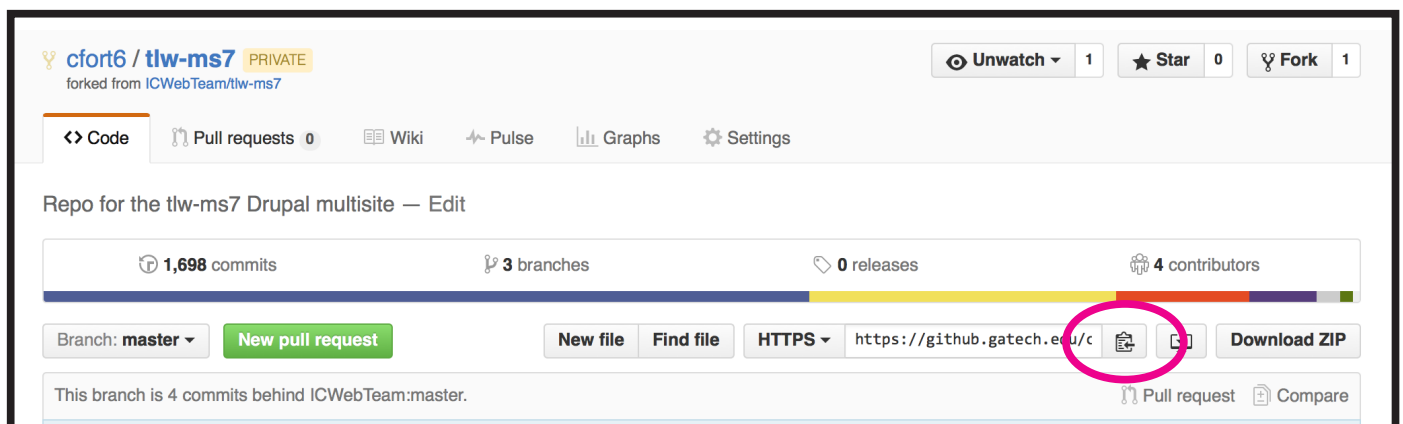
Reference: https://help.github.com/articles/fork-a-repo/

In **Github**, navigate to the repository to be forked and select "Fork" on the upper right. Select your own user name when prompted. This creates a copy of the repository for you on Github.

2. Clone the fork to your local machine.

Now you should see your fork. Copy the URL using this button:
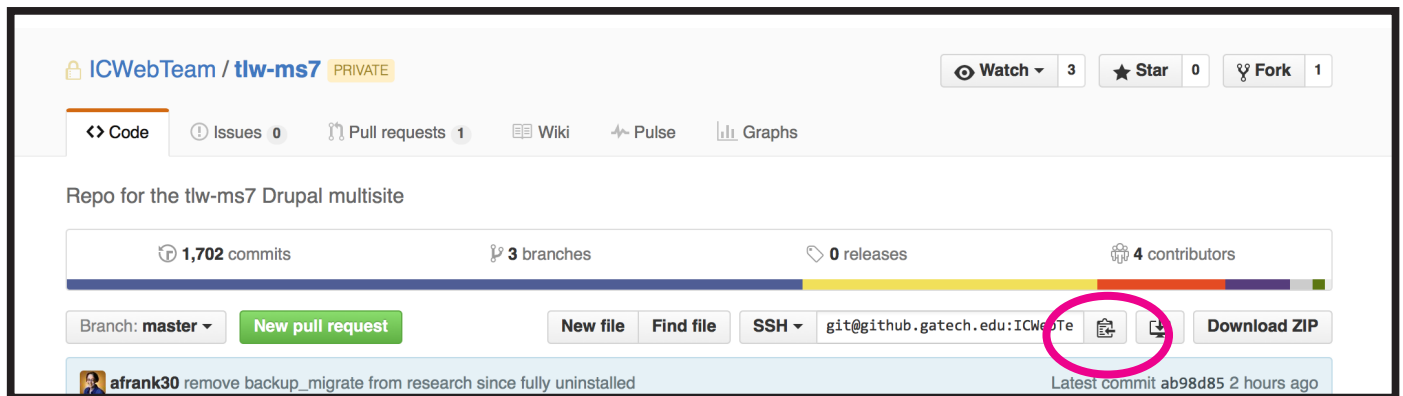
Open **Terminal** and type:
```
git clone https://github.gatech.edu/ORG_NAME/Repo-Name
```

Here you should paste the URL you copied at the beginning of this step. This creates a local copy of your clone on your machine under your current directory.

3. Sync your fork with the master.
This ensures that any changes going on with the master will be reflected in your local fork/clone.

■ Navigate in **Github** to the *original* repository. Copy the URL as shown:

■ Open **Terminal** and change directories to the fork you cloned.
type: `cd` (hit enter)
type: `ls` (look for your cloned fork/repository name)
type: `cd Fork_Name`
type: `git remote -v`

```
$ git remote -v
origin  https://github.com/YOUR_USERNAME/YOUR_FORK.git (fetch)
origin  https://github.com/YOUR_USERNAME/YOUR_FORK.git (push)
```

type: `git remote add upstream`
`https://github.gatech.edu/ORG_NAME/Repo-Name`
(All one line. This is the URL you copied at the beginning of this step)

Alternately, you can both add and fetch (sync with) your master by using this instead:
type: `git remote add -f upstream`
`https://github.gatech.edu/ORG_NAME/Repo-Name`

■ Verify your upstream repository is syncing.
   type: `git remote -v`

```
$ git remote -v
origin    https://github.com/YOUR_USERNAME/YOUR_FORK.git (fetch)
origin    https://github.com/YOUR_USERNAME/YOUR_FORK.git (push)
upstream  https://github.com/ORIGINAL_OWNER/ORIGINAL_REPOSITORY.git (fetch)
upstream  https://github.com/ORIGINAL_OWNER/ORIGINAL_REPOSITORY.git (push)
```

4. Syncing a Fork
Reference: https://help.github.com/articles/syncing-a-fork/

You probably want your fork to mirror the latest changes being made to the main repository. If you used "`git remote add -f upstream`" in the previous step, this is already done. If you used "`git remote add upstream`" and now you need to fetch upstream to sync your fork, you can use this method.

■ In **Terminal**, change to the directory of your local clone and fetch upstream to sync with the original master repository.
   type: `cd Fork_Name`
   type: `git fetch upstream`

■ Check out your fork's local master branch.
   type: `git checkout master`
   type: `git merge upstream/master`

BRANCH

5. Now Branch your issue locally.

In **Terminal**,

type: `git checkout -b name_of_your_new_branch`

COMMIT

6. Make Changes and Commit

- Change your files as needed. **Terminal**,
  type: `git status`
  (this will confirm that git is watching your modifications)

- In **Terminal**, add the files to staging (substitute your file name)
  type: `git add path/file_name`

- Then commit
  type: `git commit -m "First commit"`

You will get feedback saying how many files were changed and how.

Note: A branch may contain many commits. Name your branch to reflect what you're working on and name commits to reflect more specific changes.
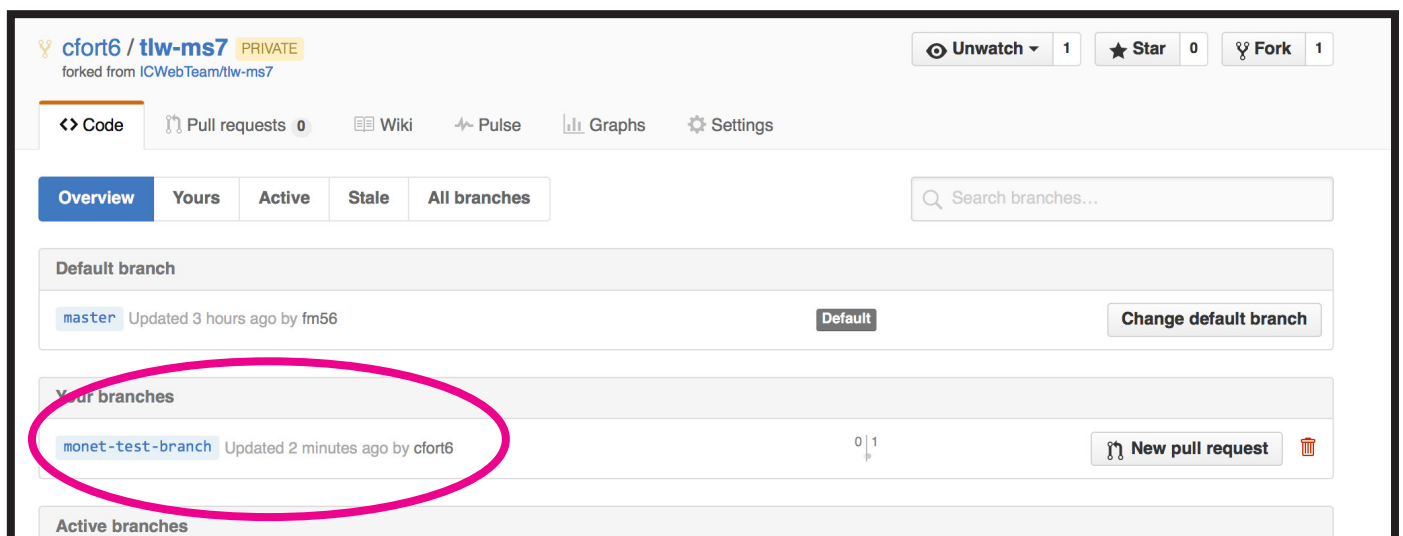
PUSH

7. Push your changes/branch to Github

- In **Terminal**,
  type: `git push origin name_of_your_new_branch`

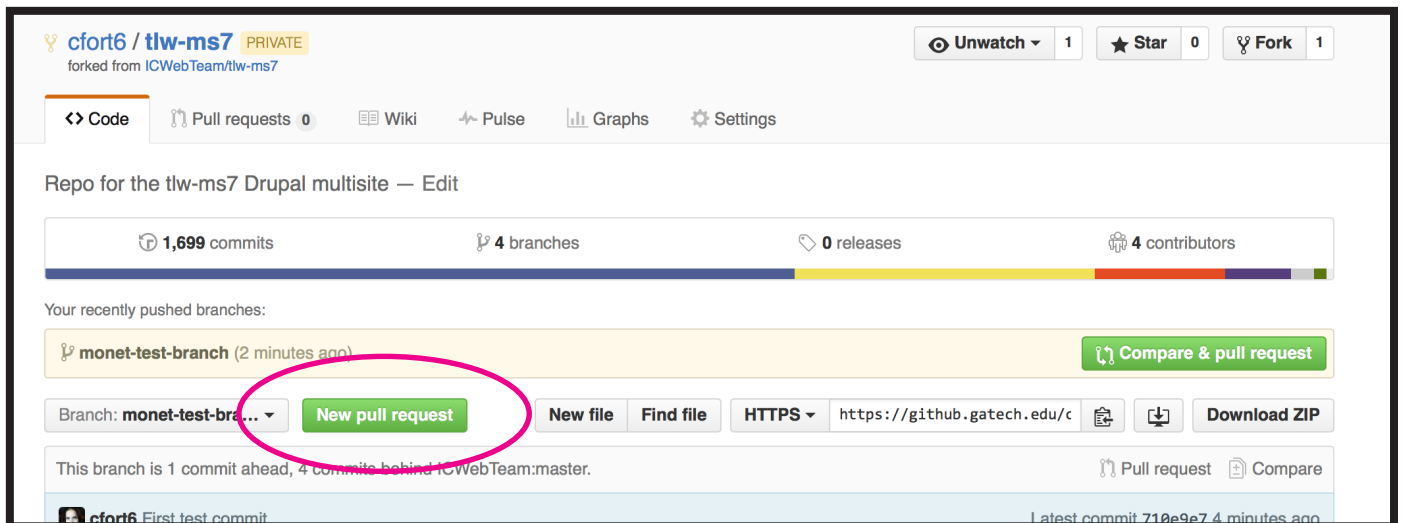- Now you should see your branch show up in **Github**.
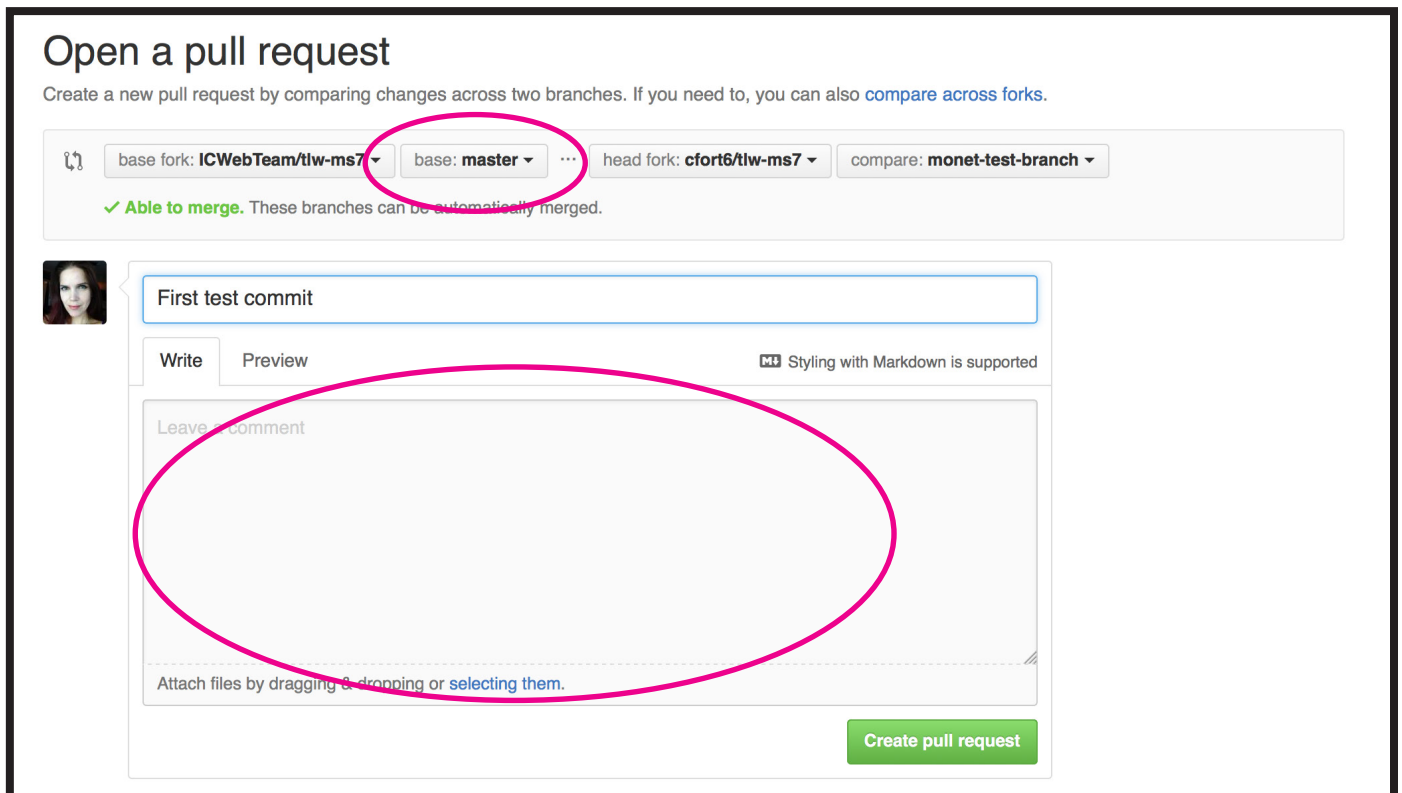
PULL REQUEST

8. New Pull Request

Reference: https://help.github.com/articles/using-pull-requests/

■ Select your branch in **Github**. Select "New pull request"

You can enter details about the changes you are suggesting.
You can also select a branch to send the request to, if it's not the master branch.

■ When you're ready, click "Create pull request" at the bottom.
After you send a pull request, any commit that's pushed to your branch will be automatically added to your pull request, which is useful if you're making additional changes.