GitHub Steps
01-11-2023

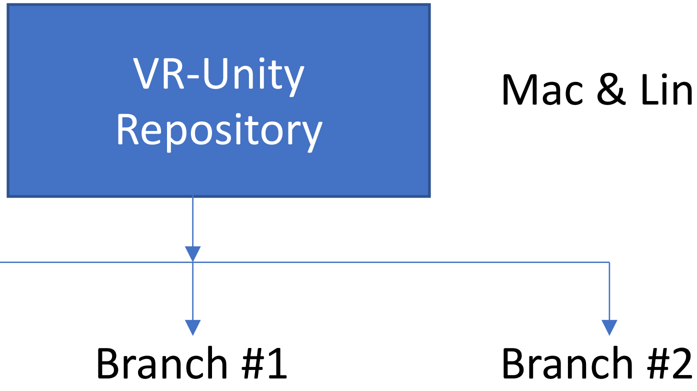
References

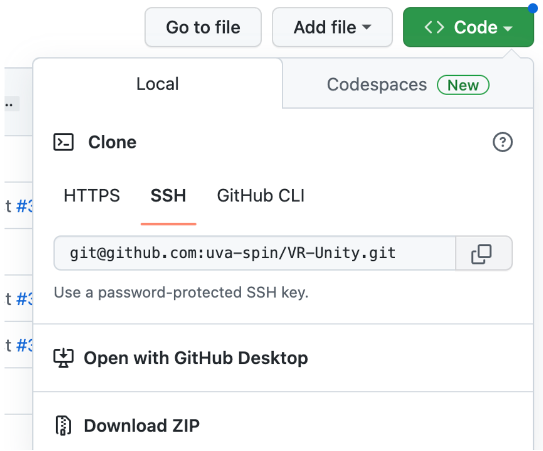
1) https://git-scm.com/book/en/v2/
2) https://jarv.is/notes/how-to-pull-request-fork-github

Some basics

Windows users:
Install Git on Windows and make sure that you have "GitBash" option available when right click

Mac & Linux users: Just use the terminal





git@github.com:uva-spin/VR-Unity.git

Use a password-protected SSH key.

➢ How to get the files?

1) Downloading the .zip file: recommended for quick/temporary tests

2) Cloning : recommended if you want to make direct changes to the mother repository
(You can do it through your fork and also through any branch)

```
$ git clone <copy>paste from the SSH tab on the repository>
```

➢ How to check where you are?

Once you cloned the repository then implement the following command

```
$ git branch –a
```

➢ How to create a branch? $ git branch <branch_name> or create and switch at the same time $ git checkout -b <branchname>

➢ How to copy a branch? $ git branch <new_branch_name> <old_branch_name>

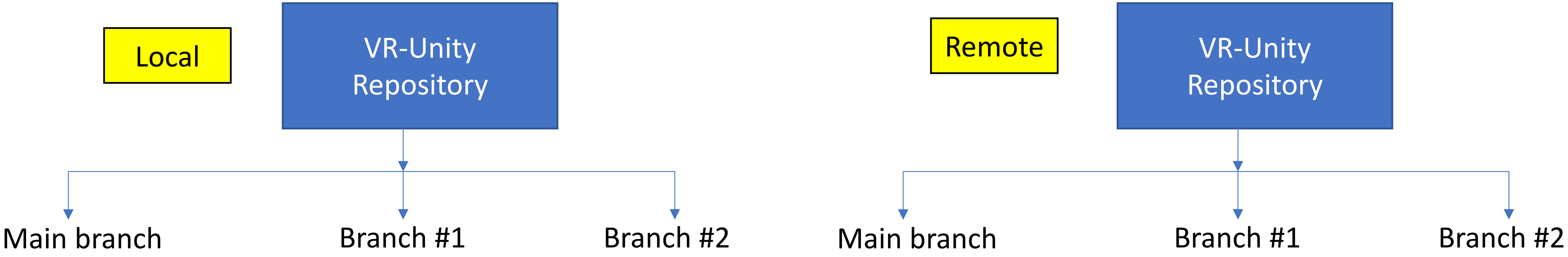
➢ How to change your branch?

```
$ git checkout <branch_name>
```

Some basics

- How to rename a branch?

```
$ git branch –m <oldname>  <newname>
```



Local

VR-Unity Repository

Main branch

Branch #1

Branch #2

Remote

VR-Unity Repository

Main branch

Branch #1

Branch #2

You can modify the files locally but will not be sync with the ”remote” repository

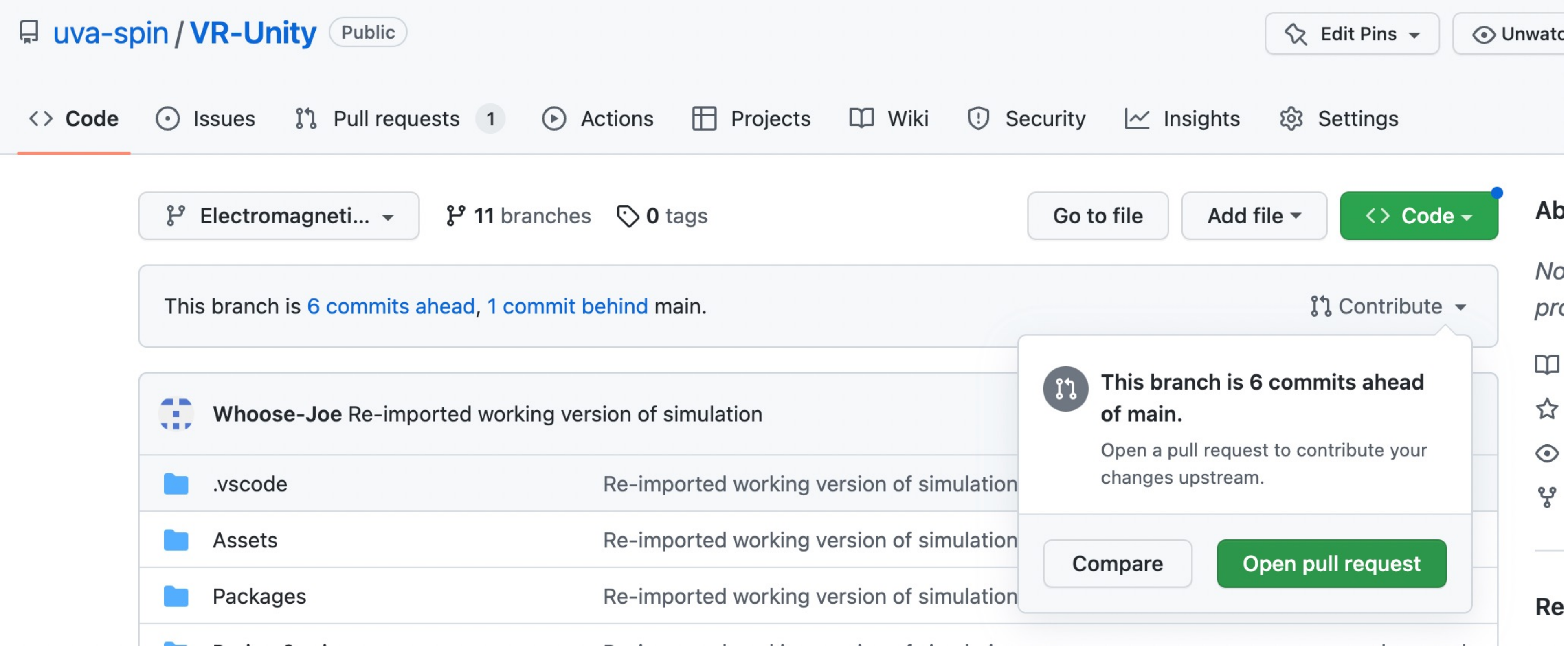
- How to check the current status of your local working branch?

```
$ git status      (I would recommend this to do be done frequently)
```

- How to update the remote branch?

```
$ git add <file_name>     (same if you deleted/modified a file)
$ git commit –m “write something meaningful about your modification”
$ git push
```

Pull Requests from a branch



This branch is 6 commits ahead, 1 commit behind main.

Whoose-Joe Re-imported working version of simulation

.vscode	Re-imported working version of simulation
Assets	Re-imported working version of simulation
Packages	Re-imported working version of simulation

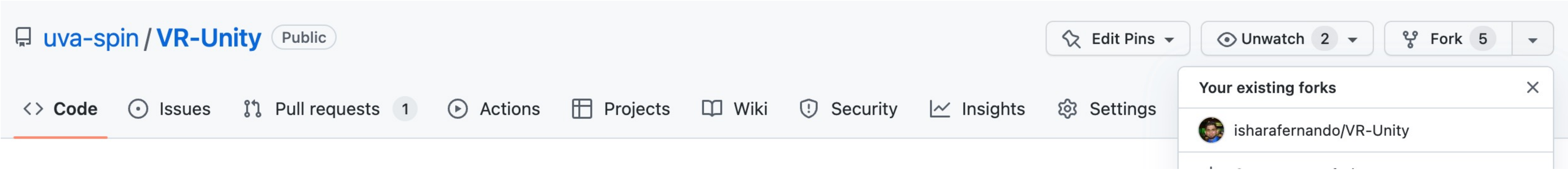
This branch is 6 commits ahead of main.

Open a pull request to contribute your changes upstream.

Once you are ready to merge to the main branch in the mother repository, then you can do it quickly via the web browser (you will have to login to your GitHub account).
If there are ”conflicts” between the main branch files and the new one, it will complain and show the conflicts. You will have to fix those before the pull-request.

Working with Forks

Creating your fork: login on your GitHub account -> Click on this “Fork” button -> Confirm



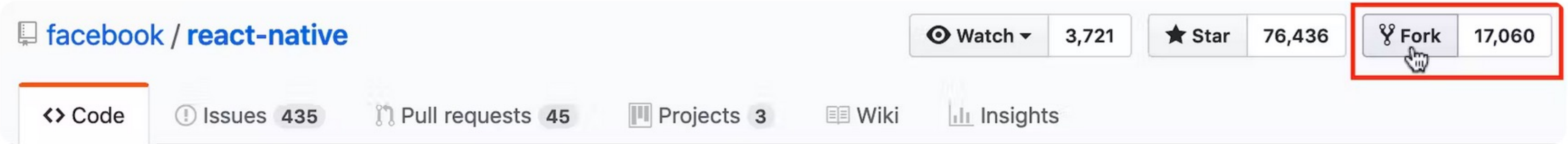
Important facts regarding this fork

- ➢ The fork (repository) is unique to you and it has its own “main” branch
- ➢ You will have to clone your fork separately (if you cloned from the mother repository) to a different location on your computer.
- ➢ The best practice is to have your forked “main” branch always sync with the “main” branch of the mother repository.
- ➢ You can have any develop branches in your fork and would not affect to the mother repository at all unless you make a pull request.
- ➢ In any branch $ git pull command is highly recommend before modifying any file.

If you are unsure, you can copy this file to a different location on your computer and copy it back to the place where the git files are located at and then implementing $ git status will show you that you updated that file(s).
i.e you can check your code without affecting either to your fork or to the mother repository.

1. Forking the Repository

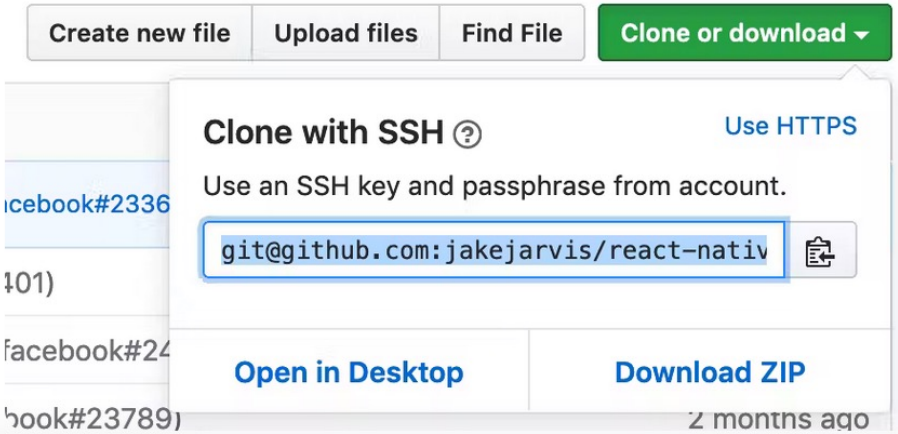
Assuming you're using GitHub, this step is easy. Just find the repository you're contributing to and press the Fork button in the upper right. This will create an exact copy of the repository (and all of its branches) under your own username.



2. Clone your new fork locally

GitHub will automatically redirect you to the forked repository under your username. This is the repository you need to clone to your local development environment, **not** the original. Grab the URL GitHub provides under the green "Clone or Download" button and plug it into the command below.

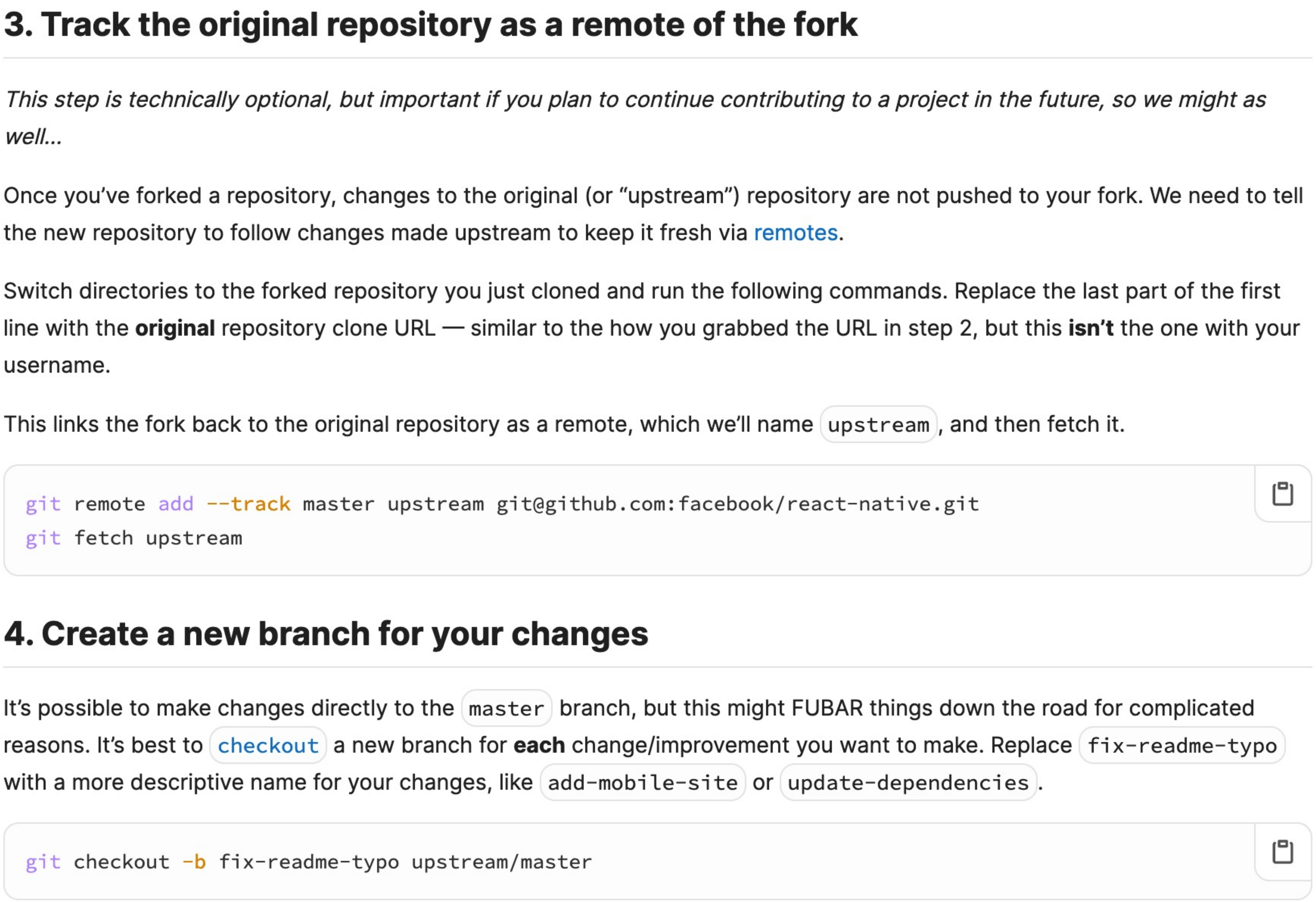
```
git clone git@github.com:jakejarvis/react-native.git
```



Use an SSH key and passphrase from account.

`git@github.com:jakejarvis/react-nativ`

Reference
https://jarv.is/notes/how-to-pull-request-fork-github/



3. Track the original repository as a remote of the fork

This step is technically optional, but important if you plan to continue contributing to a project in the future, so we might as well…

Once you've forked a repository, changes to the original (or "upstream") repository are not pushed to your fork. We need to tell the new repository to follow changes made upstream to keep it fresh via remotes.

Switch directories to the forked repository you just cloned and run the following commands. Replace the last part of the first line with the **original** repository clone URL — similar to the how you grabbed the URL in step 2, but this **isn't** the one with your username.

This links the fork back to the original repository as a remote, which we'll name `upstream`, and then fetch it.

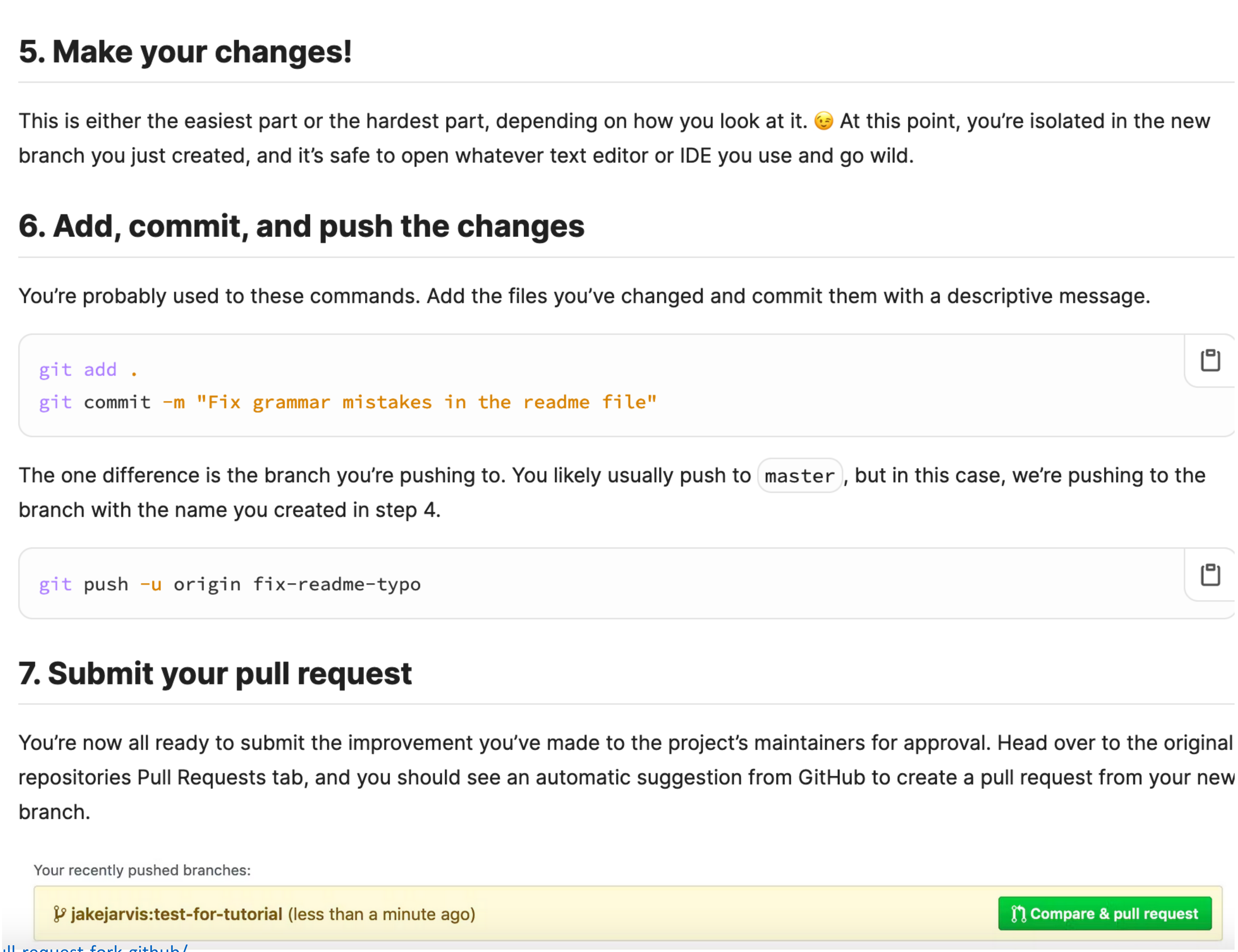
```
git remote add --track master upstream git@github.com:facebook/react-native.git
git fetch upstream
```

4. Create a new branch for your changes

It's possible to make changes directly to the `master` branch, but this might FUBAR things down the road for complicated reasons. It's best to `checkout` a new branch for **each** change/improvement you want to make. Replace `fix-readme-typo` with a more descriptive name for your changes, like `add-mobile-site` or `update-dependencies`.

```
git checkout -b fix-readme-typo upstream/master
```

Reference
https://jarv.is/notes/how-to-pull-request-fork-github/



5. Make your changes!

This is either the easiest part or the hardest part, depending on how you look at it. 😉 At this point, you're isolated in the new branch you just created, and it's safe to open whatever text editor or IDE you use and go wild.

6. Add, commit, and push the changes

You're probably used to these commands. Add the files you've changed and commit them with a descriptive message.

```
git add .
git commit -m "Fix grammar mistakes in the readme file"
```

The one difference is the branch you're pushing to. You likely usually push to `master`, but in this case, we're pushing to the branch with the name you created in step 4.

```
git push -u origin fix-readme-typo
```

7. Submit your pull request

You're now all ready to submit the improvement you've made to the project's maintainers for approval. Head over to the original repositories Pull Requests tab, and you should see an automatic suggestion from GitHub to create a pull request from your new branch.

Your recently pushed branches:

jakejarvis:test-for-tutorial (less than a minute ago)

Reference
https://jarv.is/notes/how-to-pull-request-fork-github/