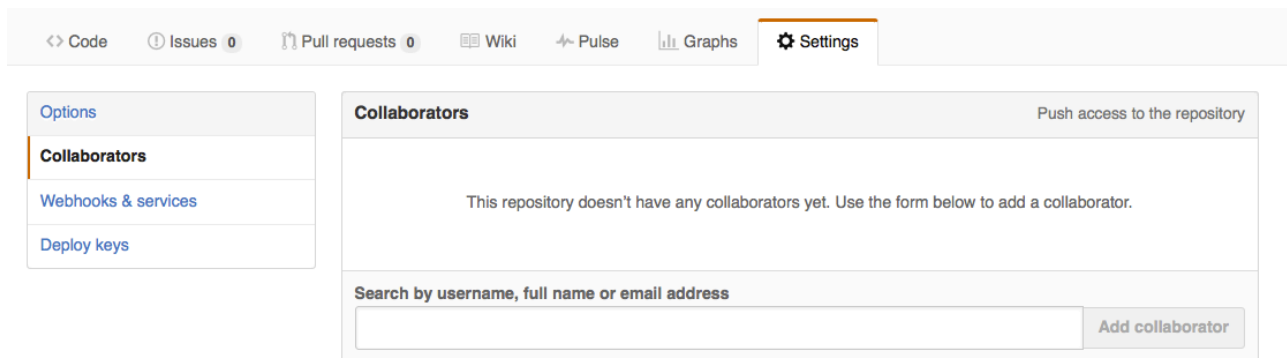


## Collaborating in Git: An Exercise

For the next step, get into pairs. One person will be the “Owner” and the other will be the “Collaborator”. The goal is for the Collaborator to add changes into the Owner’s repository. We will switch roles at the end, so both persons will be Owner and Collaborator.

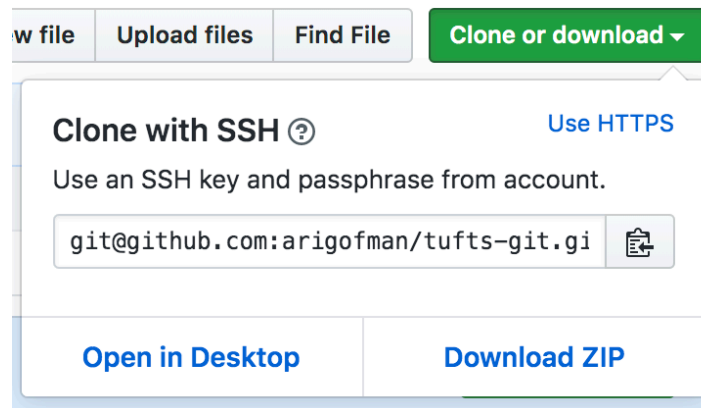
The Owner needs to give the Collaborator access to the biography repository. On GitHub, click the settings button on the right, then select Collaborators, and enter your partner’s username.



The screenshot shows the GitHub repository settings page for a repository named 'biography'. The 'Settings' tab is selected in the top navigation bar. On the left sidebar, the 'Collaborators' section is highlighted. The main content area is titled 'Collaborators' and includes a sub-header 'Push access to the repository'. Below this, a message states: 'This repository doesn't have any collaborators yet. Use the form below to add a collaborator.' There is a search input field labeled 'Search by username, full name or email address' and an 'Add collaborator' button.

To accept access to the Owner’s repo, the Collaborator should go to <https://github.com/notifications> and accept access to the Owner’s repo.

Next, the Collaborator should clone a copy of the Owner’s repository to their machine, which is similar to downloading. This is called “cloning a repo”.



The screenshot shows the 'Clone or download' dropdown menu on GitHub. The 'w file' button is selected. The dropdown menu is open, showing two options: 'Clone with SSH' (selected) and 'Use HTTPS'. The 'Clone with SSH' option includes a question mark icon and a description: 'Use an SSH key and passphrase from account.' Below this, the SSH URL 'git@github.com:arigofman/tufts-git.gi' is displayed with a copy icon. At the bottom of the dropdown, there are two buttons: 'Open in Desktop' and 'Download ZIP'.

To clone the Owner’s repo into their [Desktop](#) folder, the Collaborator should enter the URL of the other person’s repository, which you can find in the green “Clone or download” button on github. Be sure to use the SSH key version; you can switch between SSH and HTTPS in the top right corner where it says “Use HTTPS”. SSH starts with [git@github.com](https://github.com) as in this key.

```
$ git clone git@github.com:NAME/name-biography.git ~/Desktop/NAME-bio
```

IMPORTANT: replace ‘NAME’ with the Owner’s username, and ‘name’ (lowercase) with their actual name.

The Collaborator can now make a change in their clone of the Owner’s repository, the same way as we’ve been doing:

```
$ cd ~/Desktop/NAME-bio
$ cd biography
```

```
$ ls
$ nano my-biography.txt
[OWNER NAME] is learning git with [YOUR NAME]
(exit and save)
$ git add my-biography.txt
$ git commit -m "Add notes about learning git"
```

Then push the change to the *Owner's repository* on GitHub:

```
$ git push origin master
```

Note that we didn't have to create a remote called `origin`: Git uses this name by default when we clone a repository. (This is why `origin` was a good choice earlier when we were setting up remotes by hand.)

Take a look to the Owner's repository on its GitHub website now (you might need to refresh your browser.) You should be able to see the new commit made by the Collaborator.

To download the Collaborator's changes from GitHub, the Owner can now enter:

```
$ git pull origin master
```

Now the three repositories (Owner's local, Collaborator's local, and Owner's on GitHub) are back in sync.

## A Basic Collaborative Workflow

In practice, it is good to be sure that you have an updated version of the repository you are collaborating on. In this exercise, run `git pull` before making our changes. The basic collaborative workflow would be:

- update your local repo with `git pull origin master`,
- make your changes and stage them with `git add`,
- commit your changes with `git commit -m`, and
- upload the changes to GitHub with `git push origin master`

It is better to make many commits with smaller changes rather than of one commit with massive changes: small commits are easier to read and review.

Note: when working on a real project, you may want to use `git fetch` and then `git merge`. Collaboration workflows vary widely depending on your team's needs; for some ideas see <https://www.atlassian.com/git/tutorials/comparing-workflows>.

**Switch Roles and Repeat:** Switch roles and repeat the process.