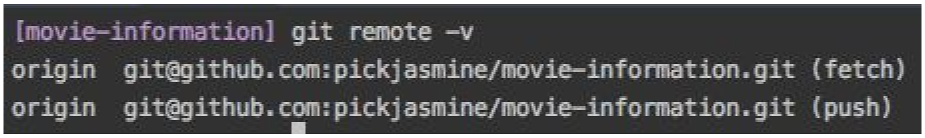
There are two common ways to collaborate through git: branching and forking.

Since everyone has already cloned the movie-information repository to their computers, here’s how to create a copy of it to your own git account so you can freely make changes and save them remotely. This technique is called **forking**. It basically saves a copy of a repository *that you now own*.

Type `git remote -v` in your terminal/command line

You should see something like this



```
[movie-information] git remote -v
origin  git@github.com:pickjasmine/movie-information.git (fetch)
origin  git@github.com:pickjasmine/movie-information.git (push)
```

Whenever you first clone a repo, that will be recognized as ‘origin’.

Go to your Github/Bitbucket/Gitlab account in your browser and create a new repository named ‘movie-information’. Do not initialize it with a README file if it asks you.

Copy the SSH of the repository you just made.

Type `git remote add <firstInitialLastName> <SSH of your repository>`
(example: Jeff would type `git remote add jgullion git@bitbucket.org:jeffgullion/movie-information.git`)

Type `git remote -v` in your terminal/command line again.

You should now see two remotes listed: origin and the new remote you just made

Make a change to a file in the movie-information repo. (Can be something super small, like adding a new line at the end of a file)

Type `git add <path/to/file.js>` to add file that you changed into your staging area.

Type `git commit -m “Setting up my own fork”` to commit your changes with the message you sent.

Type `git push -u <firstInitialLastName> master` to push to your own repository.

By adding the `-u` flag in that command, you are “setting your upstream”, AKA telling git that whenever you push or pull without specifying which remote you want, it will automatically pick what you have set as your upstream. From now on when you commit and push to your remote, you will be interacting with your own repository!