A very brief introduction to git

Jon Tjun Seng Lo Kim Lin
jlokimlin@math.ucsb.edu

UCSB Hypatian Seminar
04/25/16

You broke the build! - **Anonymous**

Outline

What is the purpose of git?

- Suppose you're an exemplary grad student, working **VERY** diligently on your research.
- As you're progressing through your project, it is natural for you to experiment.
- Then you realize, your experiment/the direction you took went awry.
- You reach a point where you say **"D#@&$....why did I not save that earlier version of my project!"**

Git comes to the rescue

- No more saving `version1.f90, version2.f90, version3.f90, version4.f90, version5.f90,...` of your code.
- Git will assist in tracking your revisions.
- More formally, Git is a **distributed version control (DVC)** and **source code management (SCM)** system.
- More importantly however, Git is your new best friend!

The basics of Git

1. Where can I get Git?
2. Intro and basic commands
3. Remote repositories
4. Resources

Where can I get Git?

Installing Git: You can either install it as a package or via another installer, or download the source code and compile it yourself.

Installing on Mac

- A Mac OS X Git installer is maintained and available for download at http://git-scm.com/download/mac, or better yet;
- Homebrew, the missing package manager for OS X, allows you to easily install hundreds of open-source tools, see: https://brew.sh/

  With Homebrew, installing Git is as easy as this:

```
brew update
brew install git
```

- For more detailed instructions, see: http://kj-prince.com/install-git-mac-osx-homebrew/.

Installing on Windows

Windows:http://msysgit.github.io/

Installing on GNU/Linux

Use the package-manager that comes with your distribution.

- If you're on a Debian-based distribution like Ubuntu, try `apt-get`:

```
$ sudo apt-get install git
```

- If you're on Fedora for instance, try `yum`:

```
$ sudo yum install git-all
```

For different Unix-like flavors, see:
http://git-scm.com/download/linux.

Basic commands

- `git init` - initializes a Git repository.
- `git add` - adds files to your repository.
- `git commit` - creates a commit. Commits allow Git to keep track of your revisions, similar to saving files.
- `git status` - shows the status of your Git repository (i.e. what files have been changed, what files are not accounted for, etc).

The basic Git workflow is

- **modify files** in your working directory.
- **stage files** you've worked on. This prepares a snapshot of the directory
- **commit the files** you've staged. This stores that snapshot in the git repository.

Initializing repositories

To initialize a new project, in the project directory, initialize the git repository with:

```
git init
```

The second way is:

```
git clone https://github.com/jlokimlin
/intro\_to\_git.git
```

Warning: Each new repository should be in its own directory. One git repository should **not** be created or cloned in an **existing git repository**, i.e., a folder you've initialized a git repository and its subfolders.

Configuring Git

You will have to do this **once** per computer you use `git config --global`:

```
git config --global user.name "Your␣Avatar"
git config --global user.email you@domain.com''
git config --global core.editor vim
git config color.ui auto
```

You can check your configuration with:

```
git config --list
```

Working locally

Right now, nothing in the project is tracked. First let's create some directories and files in our directory.

```
touch README.md
```

First, you need to tell git this file exists:

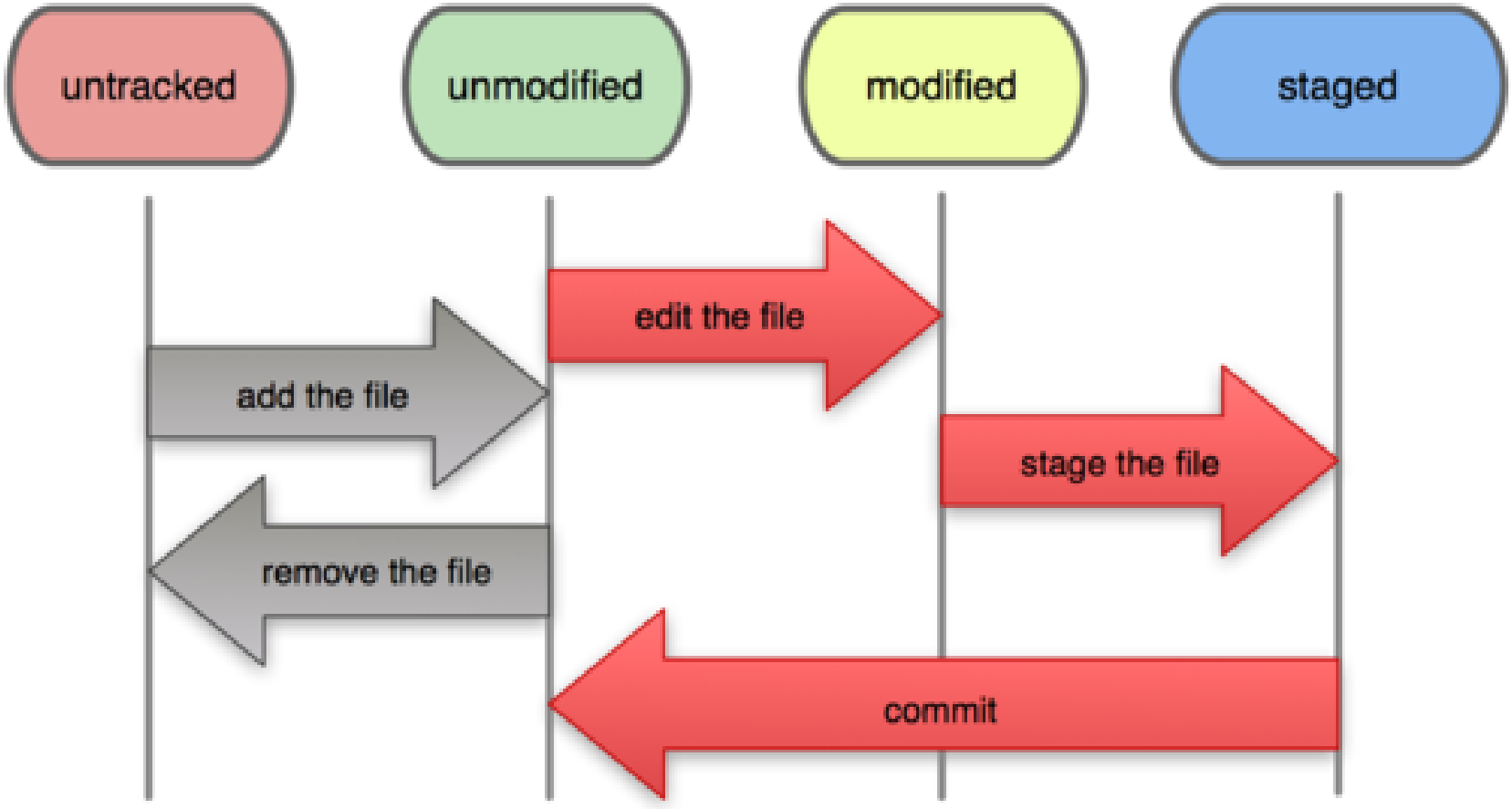
```
git add README.md
```

Now you can commit it:

```
git commit -m "Added␣a␣README␣file"
```

At this point, you have one tracked file, and an initial commit.

File Status Lifecycle



The git status command

The `git status` command will display all untracked files, and modified and staged files in your directory:

```
touch AUTHORS.txt
git status
```

Now, try

```
git add AUTHORS.txt
git status
```

`git add` is also used to stage file. In fact, running `git add` on an untracked file not only tracks it, but stages it.

```
git commit -m "Added␣the␣AUTHORS␣file"
```

And here is the second commit!

Tracking changes to a modified file

Sometimes, you may want to look at the changes you've made to a modified file:

```
git diff
```

To look at the changes you've made in the staged files, simply use:

```
git diff --cached
```

And to view the history of all commits:

```
git log
```

(Re)moving files

- Deleting files: Why use `git rm` to remove a file instead of `rm`?
  - If you just use `rm`, you will need to follow it up with `git add <fileRemoved>` The command `git rm` does both in one step.
  - You can also use `git rm --cached` which will remove the file from the index (staging it for deletion on the next commit), but keep your copy in the local file system.
- Moving files: In a similar fashion, `git mv` can be used to move a file in one step.

Canceling stages

Two scenarios may occur:

1. you staged a file you do not want to commit
2. you made some changes on a file you want to cancel.

First, let's assume you've staged a file you want to unstage:

```
touch myFile.tex
git add myFile.tex
```

To unstage it, run:

```
git reset HEAD myFile.tex
```

Second, say you've modified a file and you want to cancel the changes.

```
git checkout myFile.tex
```

Remote repositories - Main competitors

- GitHub: https://github.com/
- Bitbucket: https://bitbucket.org/

Create your own remote repository

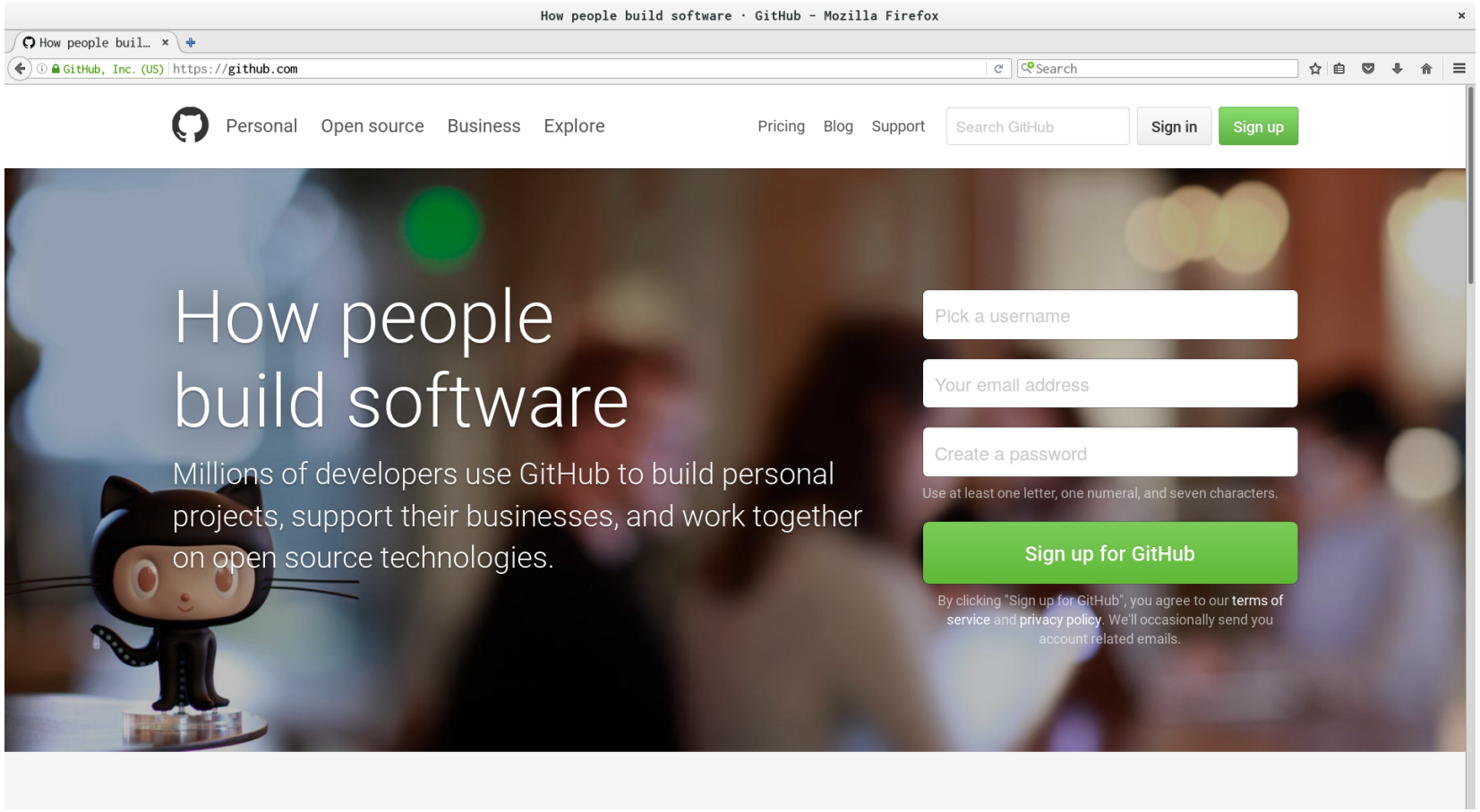
Create your own repository on GitHub ($7 a month for private repo's) or Bitbucket (free unlimited private repo's).

This can be for an existing or a new project.

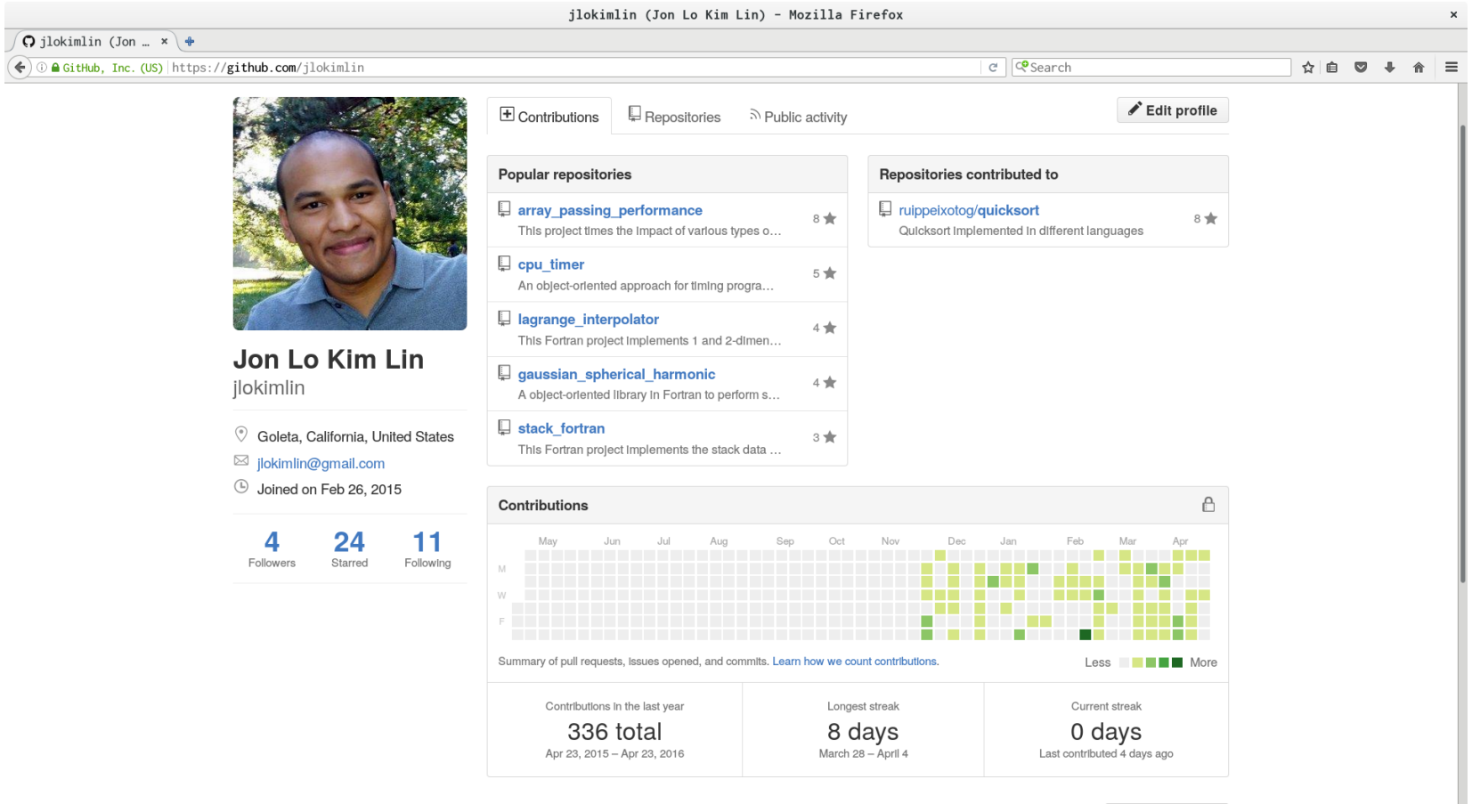
More commands:

- `git clone` - imports a remote repository.
- `git pull` - extracts most recent changes from a repository.
- `git push` - broadcast your changes to a repository.

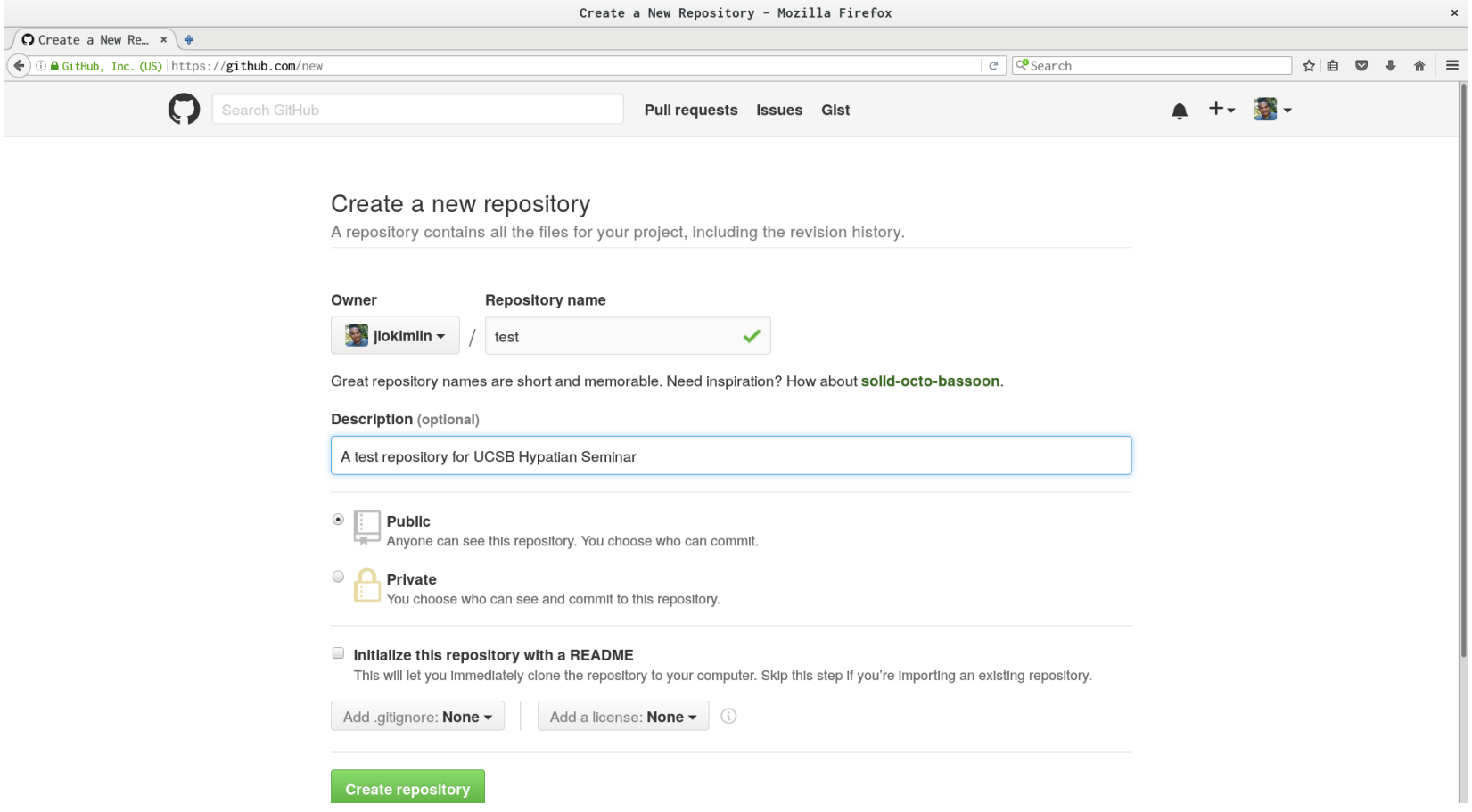
A distant repository with GitHub



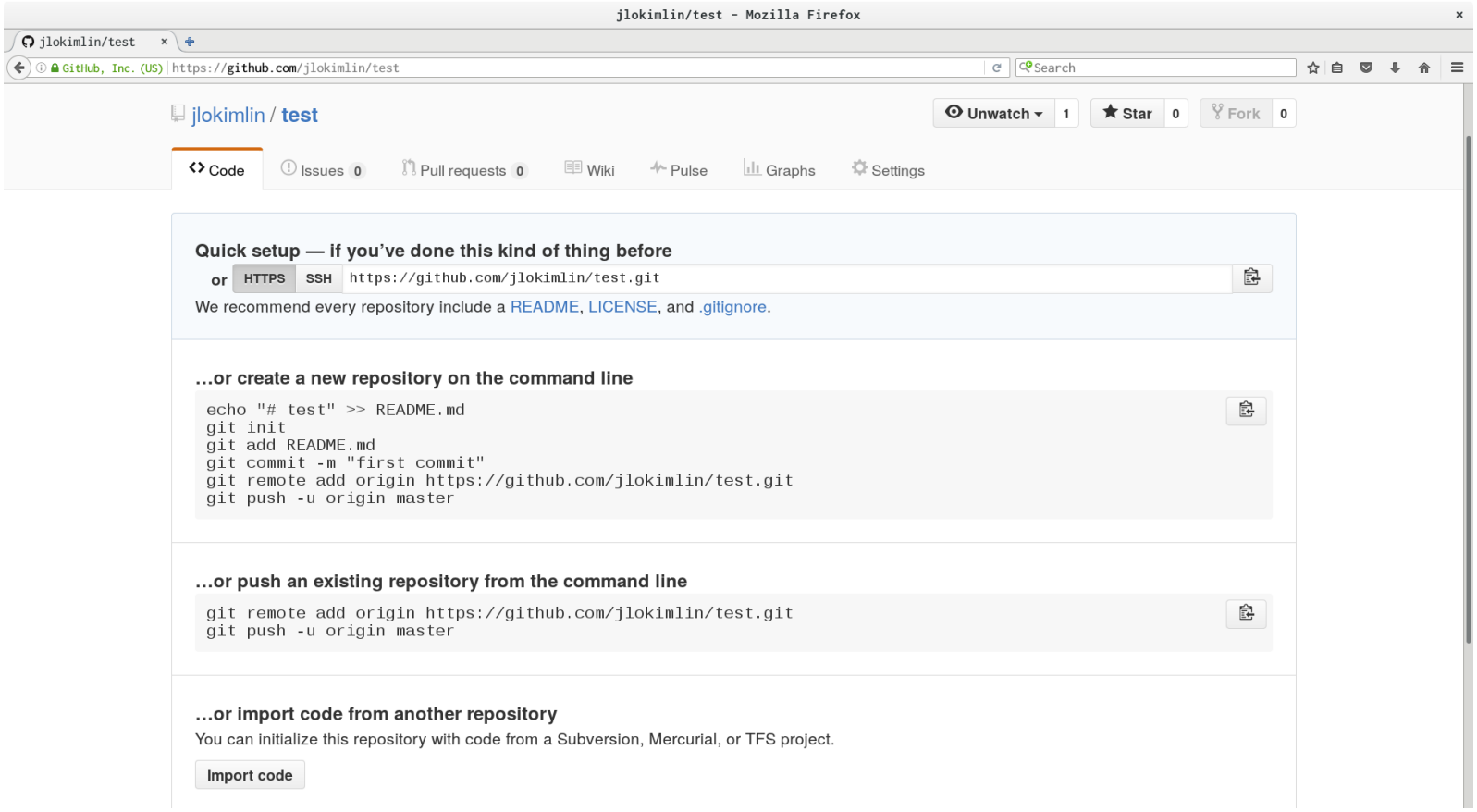
GitHub profile



Create new repository



Create a new repository (continued)



Create a new repository on the command line

```
mkdir /path/to/your/project
cd /path/to/your/project
echo "# test" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin
https://github.com/jlokimlin/test.git
git push -u origin master
```

Push an existing repository from the command line

```
mkdir /path/to/your/project
cd /path/to/your/project
git remote add origin
https://github.com/jlokimlin/test.git
git push -u origin master
```

Resources

- A personal favorite on Bitbucket for git: https://www.youtube.com/watch?v=BtEvnE79jxY
- Everyday git with 20 commands: https://www.kernel.org/pub/software/scm/git/docs/everyday.html
- Git tutorial: https://www.kernel.org/pub/software/scm/git/docs/gittutorial.html
- The (overwhelming) Git manual page: https://www.kernel.org/pub/software/scm/git/docs/

Thank you for your time