

Bioinformatics Lessons Schedule

Date	Subject
10-22	Terminus
10-29	Server Basics
11-5	Server Basics, continued
11-12	Server Basics, continued
11-19	Server Basics, continued
11-26	No lesson, week of Thanksgiving
12-03	Server Basics, repeat of Week 5, continued
12-10	Skipped for symposium
12-17	Basic Git and GitHub
12-24	Christmas break
12-31	Christmas break
01-07	Resume lessons with RNA-seq

Git and GitHub

What is Git?

THIS IS GIT. IT TRACKS COLLABORATIVE WORK
ON PROJECTS THROUGH A BEAUTIFUL
DISTRIBUTED GRAPH THEORY TREE MODEL.

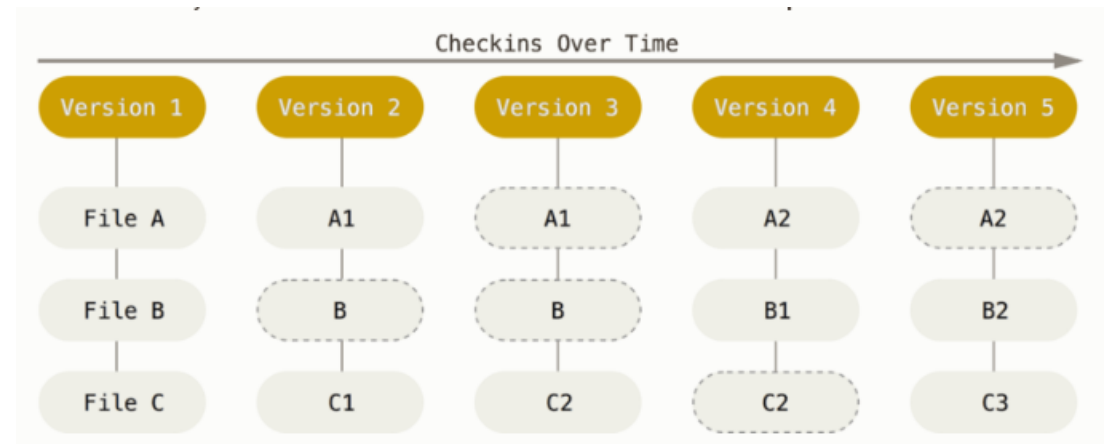
COOL. HOW DO WE USE IT?

NO IDEA. JUST MEMORIZE THESE SHELL
COMMANDS AND TYPE THEM TO SYNC UP.
IF YOU GET ERRORS, SAVE YOUR WORK
ELSEWHERE, DELETE THE PROJECT,
AND DOWNLOAD A FRESH COPY.



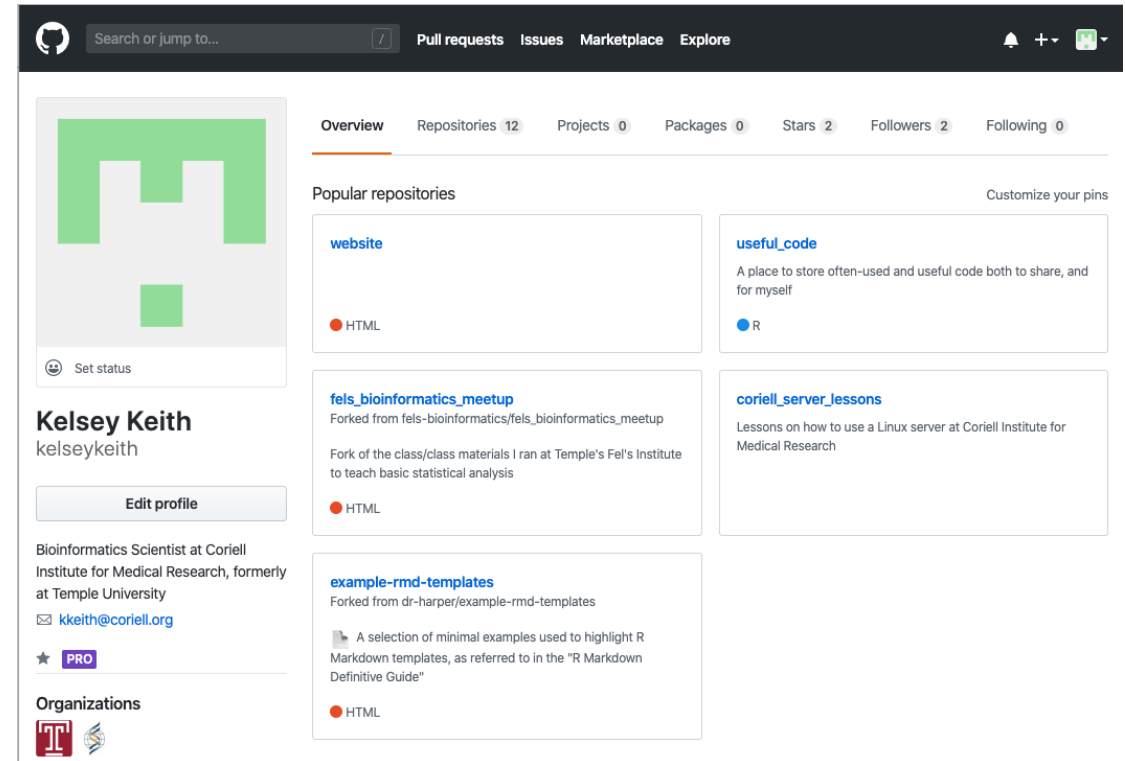
Git is a Version Control System

- Git works as a series of snapshots of a file system
- Originally made for collaborative development of the Linux operating system
- Every time you save your project through Git, Git takes a picture of it, with links between the current and past versions of files
- If you need to undo a change you can go back to previous versions of the file



What do we use Git for?

- Documenting a project's progress
- Working with other people
- Reproducible research
- Skills advertisement (for the future)



The screenshot shows a GitHub profile page for Kelsey Keith. The profile includes a green and white avatar, a bio identifying her as a Bioinformatics Scientist at Coriell Institute for Medical Research, and a list of popular repositories. The repositories listed are 'website', 'fcls_bioinformatics_meetup', 'example-rmd-templates', 'useful_code', and 'coriell_server_lessons'. The page also shows navigation links for Pull requests, Issues, Marketplace, and Explore, and a search bar at the top.

Getting Started

Set Up Your `.gitconfig` Files

Set Up Your `.gitconfig` Files

1. Log onto the server

Set Up Your `.gitconfig` Files

1. Log onto the server
2. Set your username and email address

```
git config --global user.name "Your Name"
```

```
git config --global user.email youremail@coriell.org
```

Set Up Your `.gitconfig` Files

1. Log onto the server

2. Set your username and email address

```
git config --global user.name "Your Name"
```

```
git config --global user.email youremail@coriell.org
```

3. Set your default text editor

```
git config --global core.editor nano
```

Set Up Your `.gitconfig` Files

1. Log onto the server

2. Set your username and email address

```
git config --global user.name "Your Name"
```

```
git config --global user.email youremail@coriell.org
```

3. Set your default text editor

```
git config --global core.editor nano
```

4. Check your settings

```
git config --list
```

Set Up Your SSH Key

Set Up Your SSH Key

1. Generate an SSH key for yourself

- **Type:** `ssh-keygen -t rsa -b 4096 -C "youremail@coriell.org"`
- When you're prompted to "Enter a file in which to save the key", press Enter to accept the default file location.
- At the prompt, "Enter passphrase" hit Enter to save the SSH key without a password (although you can enter a password) if you really want to.

Set Up Your SSH Key

1. Generate an SSH key for yourself

- Type:

```
ssh-keygen -t rsa -b 4096 -C "youremail@coriell.org"
```

- When you're prompted to "Enter a file in which to save the key", press Enter to accept the default file location.
- At the prompt, "Enter passphrase" hit Enter to save the SSH key without a password (although you can enter a password) if you really want to.

2. Add your SSH key to the ssh-agent

1. Start the ssh-agent in the background by entering:

```
eval "$(ssh-agent -s)"
```

2. Add your SSH key to the ssh-agent by entering:

```
ssh-add ~/.ssh/id_rsa
```

Add Your SSH Key to Your GitHub Account

1. Copy your ssh key to your clipboard

```
pbcopy < ~/.ssh/id_rsa.pub
```

2. Follow along with me as I show you how to add the key to your GitHub account.

Make Your First Repository

Initializing a Git Repository

1. Log onto the server

Initializing a Git Repository

1. Log onto the server
2. Go to your data directory
`cd data`

Initializing a Git Repository

1. Log onto the server
2. Go to your data directory
`cd data`
3. Make a new directory to practice with:
`mkdir practice_repository`

Initializing a Git Repository

1. Log onto the server
2. Go to your data directory
`cd data`
3. Make a new directory to practice with:
`mkdir practice_repository`
4. Make a git repository
`git init`

Making Your First Commit

Making Your First Commit

1. Make a file to commit

1. `nano README.md`

2. Add something to the file and save it

Making Your First Commit

1. Make a file to commit

1. `nano README.md`

2. Add something to the file and save it

2. Check your repository's status

`git status`

Making Your First Commit

1. Make a file to commit

1. `nano README.md`

2. Add something to the file and save it

2. Check your repository's status

`git status`

3. Add the file(s)

`git add README.md`

Making Your First Commit

1. Make a file to commit

1. `nano README.md`
2. Add something to the file and save it

2. Check your repository's status

```
git status
```

3. Add the file(s)

```
git add README.md
```

4. Commit

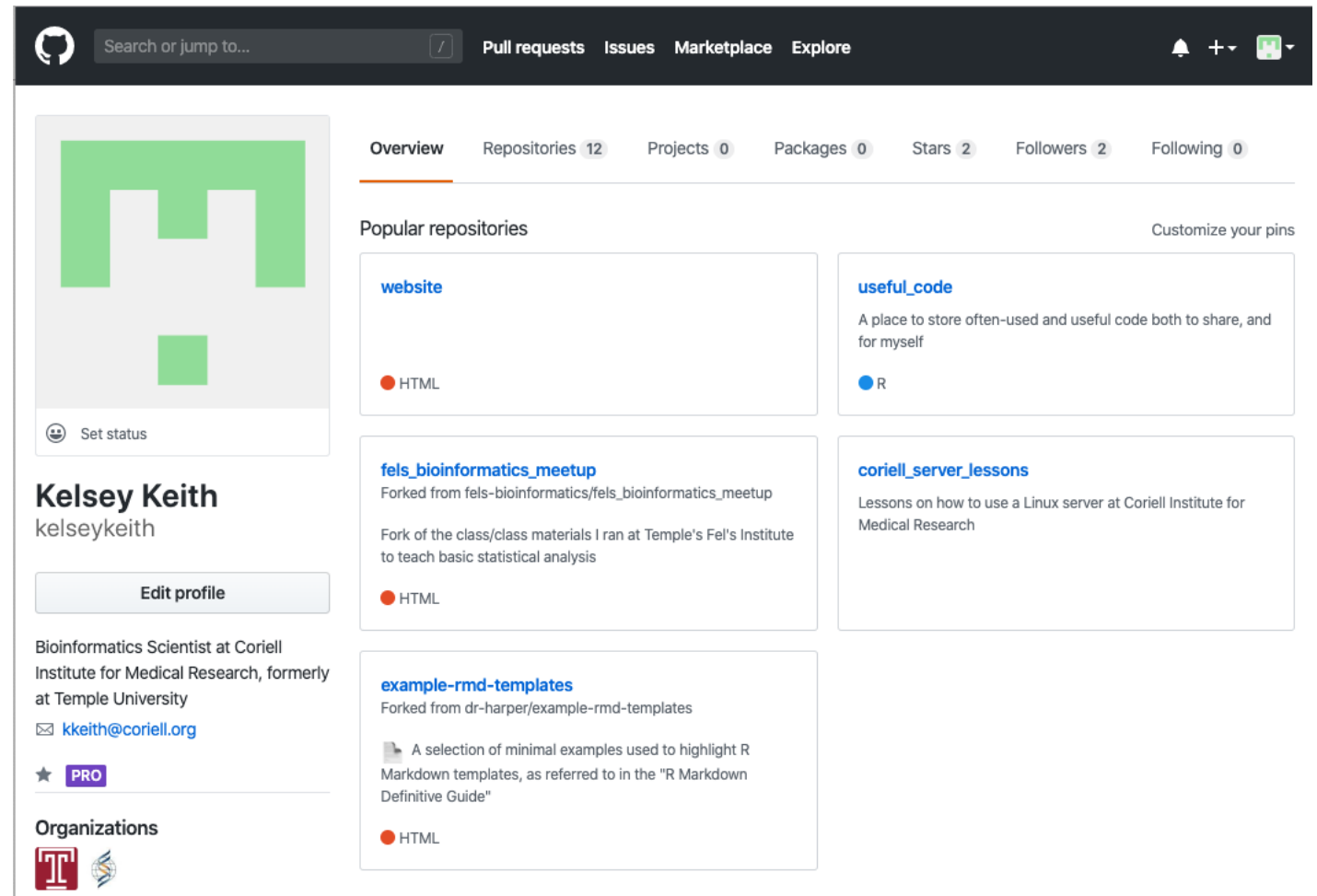
```
git commit -m 'first commit'
```

Connecting Your Local Repository to GitHub

1. Create a repository on GitHub to server as the remote

Connecting Your Local Repository to GitHub

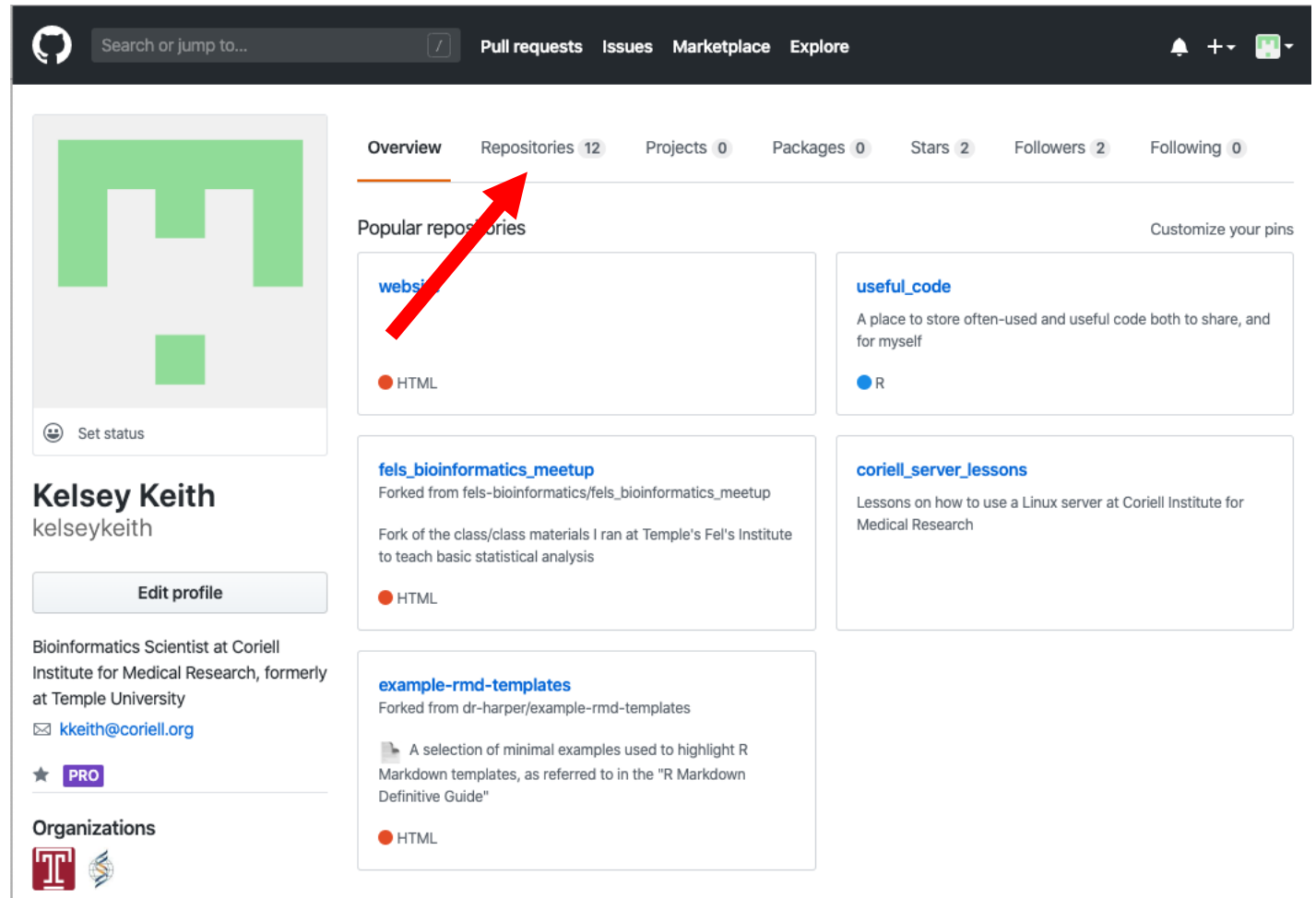
1. Create a repository on GitHub to server as the remote



The screenshot shows the GitHub profile page for Kelsey Keith. The profile includes a bio: "Bioinformatics Scientist at Coriell Institute for Medical Research, formerly at Temple University" and an email address "kkeith@coriell.org". The user has a PRO membership. The "Organizations" section shows logos for Temple University and Coriell Institute. The "Popular repositories" section lists three repositories: "website" (HTML), "fcls_bioinformatics_meetup" (HTML, forked from fels-bioinformatics/fcls_bioinformatics_meetup), and "example-rmd-templates" (HTML, forked from dr-harper/example-rmd-templates). The "useful_code" repository is also visible, described as a place to store often-used and useful code.

Connecting Your Local Repository to GitHub

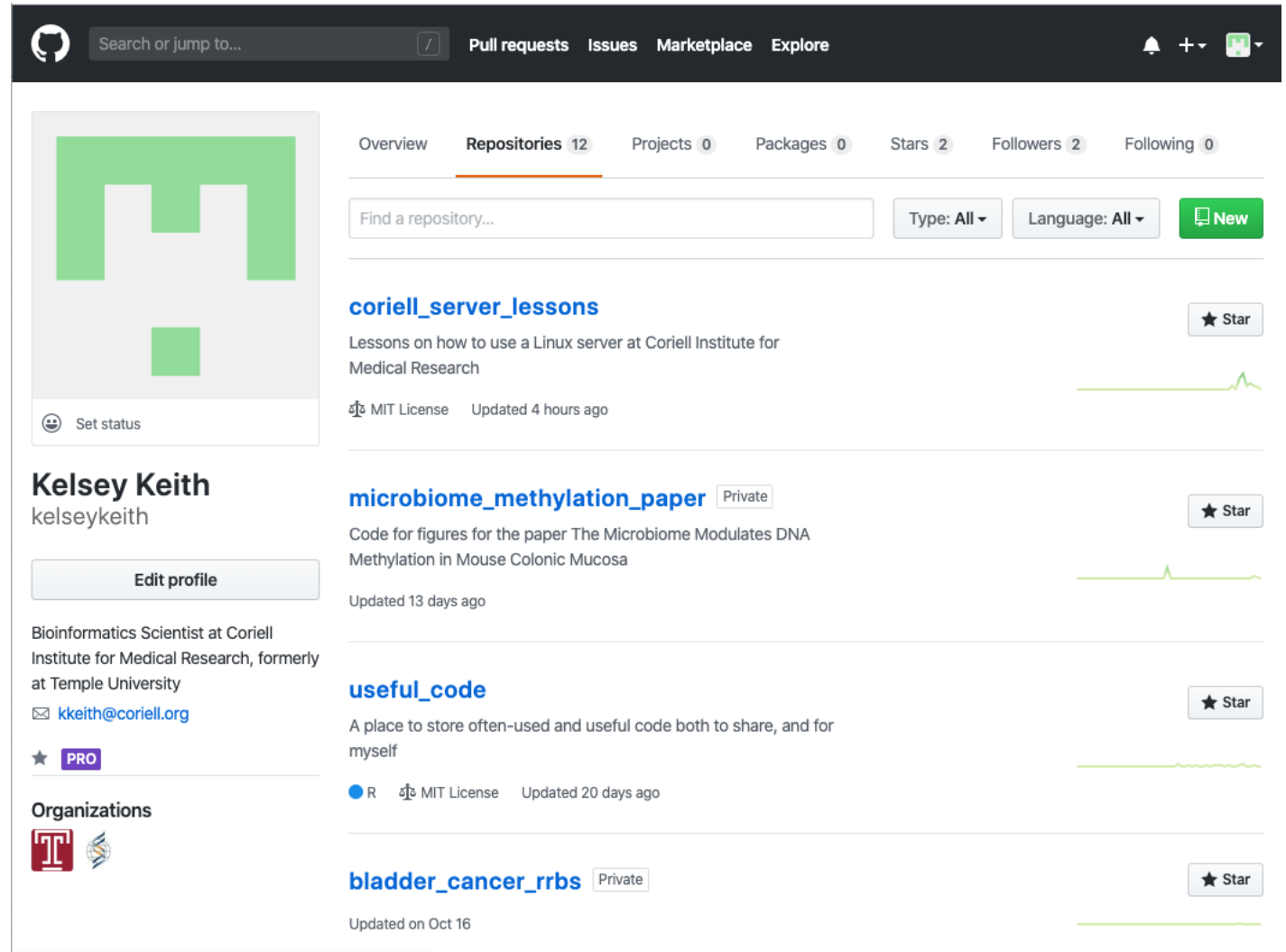
1. Create a repository on GitHub to server as the remote



The screenshot displays the GitHub profile of Kelsey Keith. The navigation bar at the top includes 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. The profile section shows the user's name, username 'kelseykeith', and a bio: 'Bioinformatics Scientist at Coriell Institute for Medical Research, formerly at Temple University'. Below the bio is an 'Edit profile' button and a 'PRO' badge. The 'Organizations' section shows logos for 'IT' and another organization. The main content area features a 'Repositories' tab with a count of 12, which is highlighted by a red arrow. Below this are several repository cards, including 'useful_code', 'fcls_bioinformatics_meetup', and 'example-rmd-templates', each with a brief description and a language indicator (HTML or R).

Connecting Your Local Repository to GitHub

1. Create a repository on GitHub to server as the remote

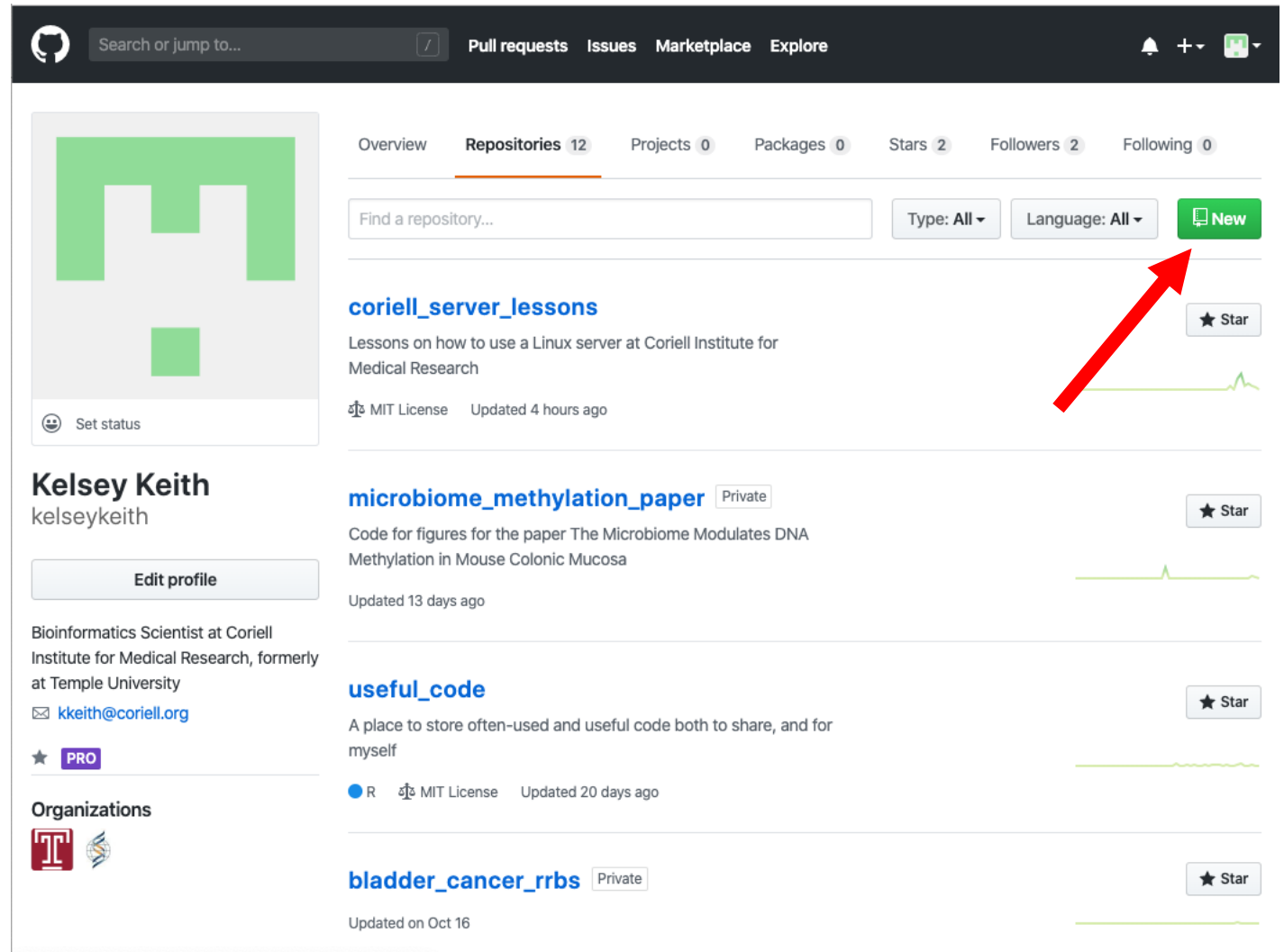


The screenshot displays the GitHub profile of Kelsey Keith. The profile includes a bio: "Bioinformatics Scientist at Coriell Institute for Medical Research, formerly at Temple University", with an email address "k Keith@coriell.org" and a "PRO" badge. The "Organizations" section shows logos for Coriell and Temple University. The "Repositories" section lists three repositories:

- coriell_server_lessons**: "Lessons on how to use a Linux server at Coriell Institute for Medical Research", MIT License, updated 4 hours ago.
- microbiome_methylation_paper** (Private): "Code for figures for the paper The Microbiome Modulates DNA Methylation in Mouse Colonic Mucosa", updated 13 days ago.
- useful_code**: "A place to store often-used and useful code both to share, and for myself", MIT License, updated 20 days ago.
- bladder_cancer_rrbs** (Private): Updated on Oct 16.

Connecting Your Local Repository to GitHub

1. Create a repository on GitHub to server as the remote

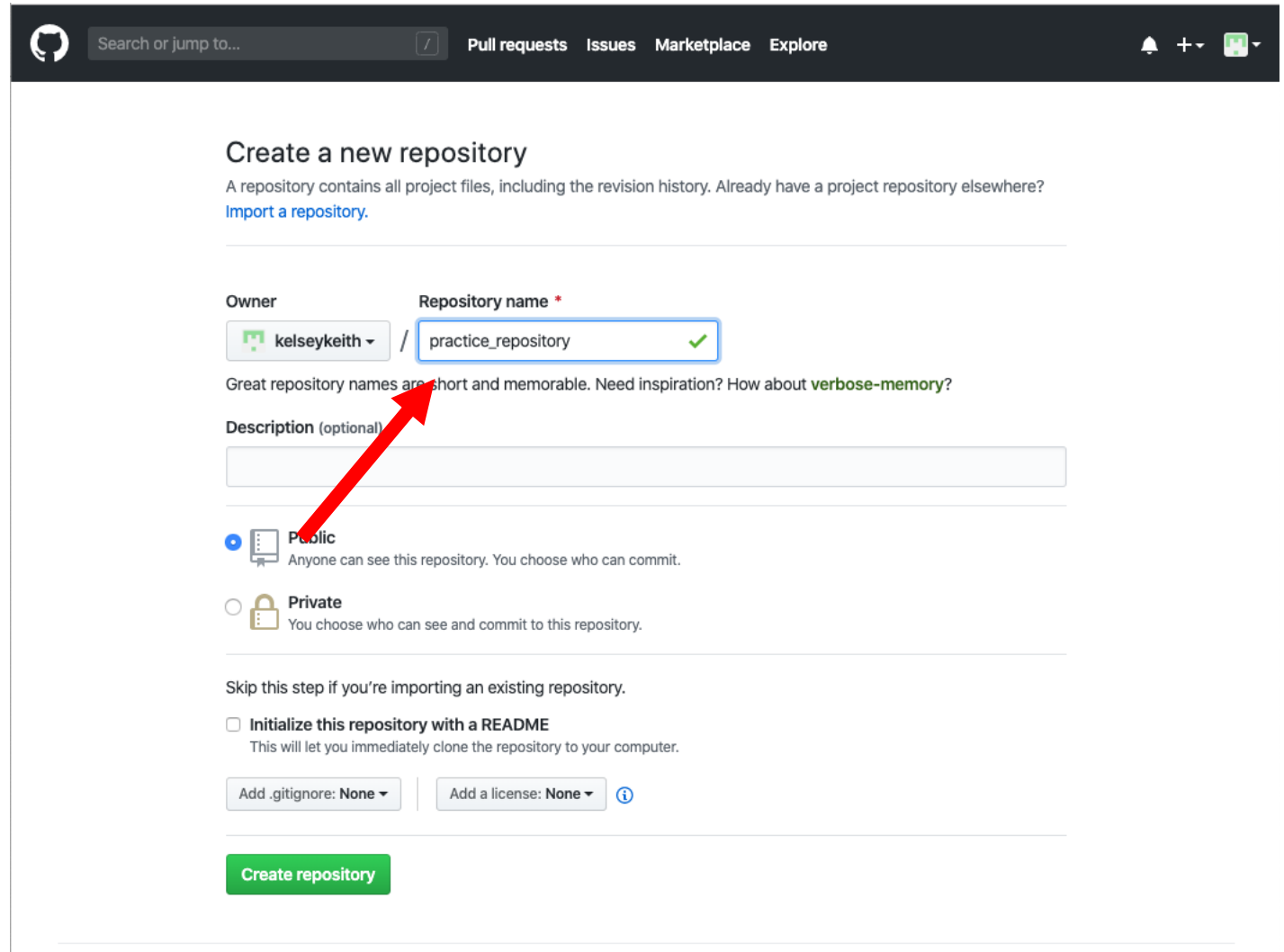


The screenshot shows the GitHub profile page for Kelsey Keith (kelseykeith). The 'Repositories' tab is active, displaying a list of repositories. A red arrow points to the 'New' button in the top right corner of the repository list. The repositories listed are:

- coriell_server_lessons**: Lessons on how to use a Linux server at Coriell Institute for Medical Research. MIT License. Updated 4 hours ago.
- microbiome_methylation_paper** (Private): Code for figures for the paper The Microbiome Modulates DNA Methylation in Mouse Colonic Mucosa. Updated 13 days ago.
- useful_code**: A place to store often-used and useful code both to share, and for myself. R. MIT License. Updated 20 days ago.
- bladder_cancer_rrbs** (Private): Updated on Oct 16.

Connecting Your Local Repository to GitHub

1. Create a repository on GitHub to server as the remote



Search or jump to... Pull requests Issues Marketplace Explore

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?
[Import a repository.](#)

Owner: kelseykeith / Repository name: practice_repository ✓

Great repository names are short and memorable. Need inspiration? How about [verbose-memory](#)?

Description (optional)

Public
Anyone can see this repository. You choose who can commit.

Private
You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

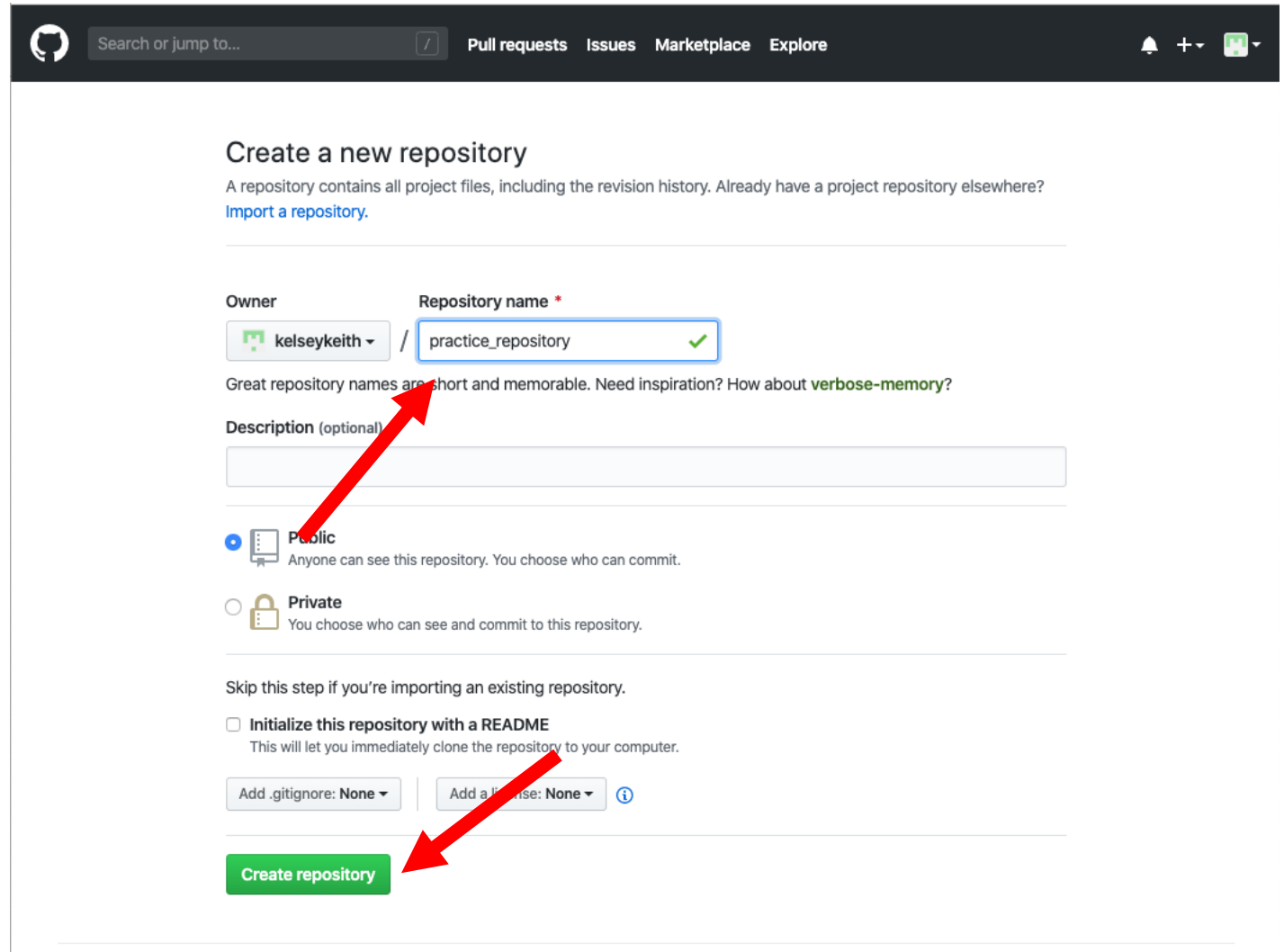
Initialize this repository with a README
This will let you immediately clone the repository to your computer.

Add .gitignore: None | Add a license: None ⓘ

[Create repository](#)

Connecting Your Local Repository to GitHub

1. Create a repository on GitHub to server as the remote



Search or jump to... Pull requests Issues Marketplace Explore

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?
[Import a repository.](#)

Owner: kelseykeith / Repository name: practice_repository ✓

Great repository names are short and memorable. Need inspiration? How about [verbose-memory](#)?

Description (optional)

Public
Anyone can see this repository. You choose who can commit.

Private
You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

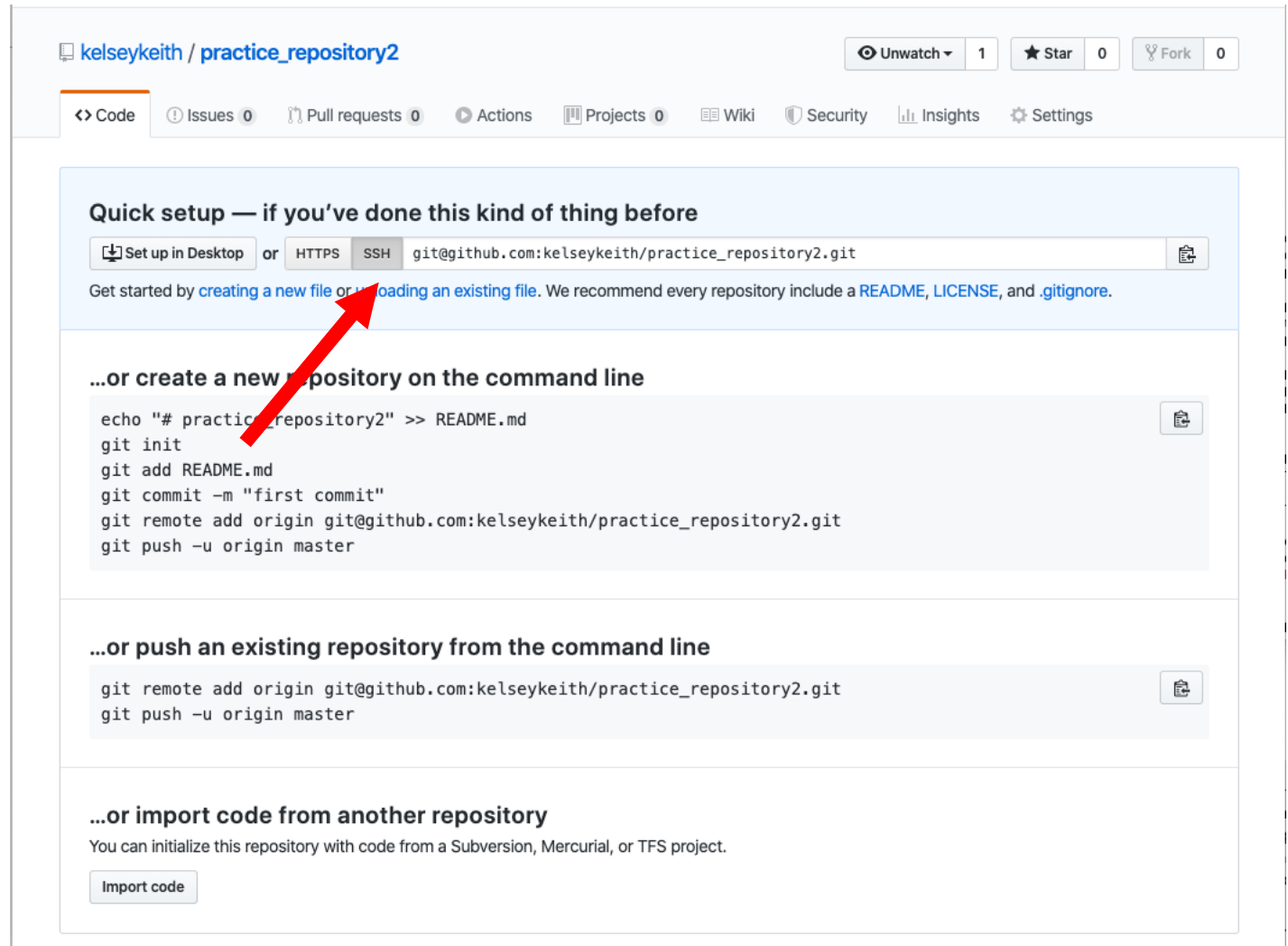
Initialize this repository with a README
This will let you immediately clone the repository to your computer.

Add .gitignore: None | Add a license: None ⓘ

Create repository

Connecting Your Local Repository to GitHub

1. Create a repository on GitHub to server as the remote



The screenshot shows the GitHub interface for a repository named 'practice_repository2' by user 'kelseykeith'. The repository has 1 Unwatch, 0 Stars, and 0 Forks. The navigation bar includes links for Code, Issues (0), Pull requests (0), Actions, Projects (0), Wiki, Security, Insights, and Settings. The main content area is titled 'Quick setup — if you've done this kind of thing before' and provides three options for cloning the repository:

- Set up in Desktop** or **HTTPS** **SSH** `git@github.com:kelseykeith/practice_repository2.git`
- ...or create a new repository on the command line**

```
echo "# practice_repository2" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin git@github.com:kelseykeith/practice_repository2.git
git push -u origin master
```
- ...or push an existing repository from the command line**

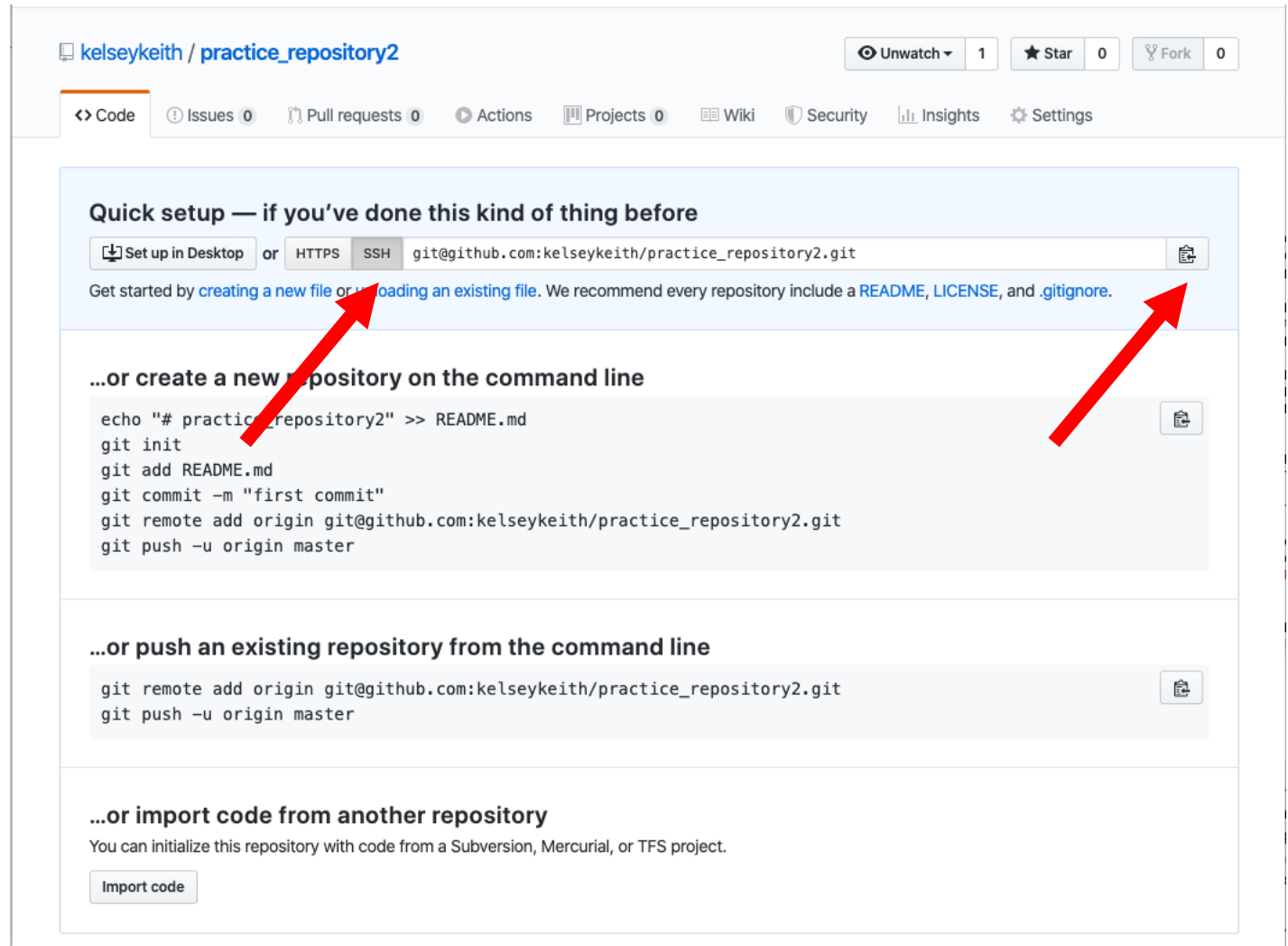
```
git remote add origin git@github.com:kelseykeith/practice_repository2.git
git push -u origin master
```

Below these options, there is a section for **...or import code from another repository** with a note: 'You can initialize this repository with code from a Subversion, Mercurial, or TFS project.' and an 'Import code' button.

A red arrow points to the 'SSH' option in the first section.

Connecting Your Local Repository to GitHub

1. Create a repository on GitHub to server as the remote



The screenshot shows the GitHub interface for a repository named 'kelseykeith / practice_repository2'. At the top, there are navigation tabs for 'Code', 'Issues', 'Pull requests', 'Actions', 'Projects', 'Wiki', 'Security', 'Insights', and 'Settings'. Below the repository name, there are statistics for 'Unwatch' (1), 'Star' (0), and 'Fork' (0). The main content area is divided into sections for quick setup and command-line instructions. Two red arrows point to the SSH URL and the command-line instructions.

Quick setup — if you've done this kind of thing before

Set up in Desktop or **HTTPS** **SSH** `git@github.com:kelseykeith/practice_repository2.git`

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```
echo "# practice_repository2" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin git@github.com:kelseykeith/practice_repository2.git
git push -u origin master
```

...or push an existing repository from the command line

```
git remote add origin git@github.com:kelseykeith/practice_repository2.git
git push -u origin master
```

...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

[Import code](#)

Connecting Your Local Repository to GitHub

1. Create a repository on GitHub to server as the remote
2. Add the GitHub repository as a remote

```
git remote add origin  
git@github.com:kelseykeith/practice_repository.git
```

Connecting Your Local Repository to GitHub

1. Create a repository on GitHub to server as the remote

2. Add the GitHub repository as a remote

```
git remote add origin  
git@github.com:kelseykeith/practice_repository.git
```

3. Push everything to the remote

```
git push -u origin master
```

Working with GitHub

Working with the GitHub

Working with the GitHub

1. Make an edit to your README on GitHub

Working with the GitHub

1. Make an edit to your README on GitHub
2. Use the command `git pull` on the server to grab the changes.

Working with the GitHub

1. Make an edit to your README on GitHub
2. Use the command `git pull` on the server to grab the changes.
3. Make a change on the server

Working with the GitHub

1. Make an edit to your README on GitHub
2. Use the command `git pull` on the server to grab the changes.
3. Make a change on the server
4. Add, commit, and push the change to GitHub.
 1. `git status`
 2. `git add README.md`
 3. `git commit -m 'made a practice change'`
 4. `git push`

Adding a `.gitignore`

Adding a `.gitignore`

1. What is a `.gitignore`?

- A `.gitignore` is a file that tells git to ignore stuff; it can be certain files types or entire folder or one specific file.

Adding a `.gitignore`

1. What is a `.gitignore`?

- A `.gitignore` is a file that tells git to ignore stuff; it can be certain files types or entire folder or one specific file.
- **WARNING:** The file upload limit for GitHub is **100 Mb**. Do NOT upload data to GitHub. We are using GitHub for documentation purposes only.

Adding a `.gitignore`

1. What is a `.gitignore`?

- A `.gitignore` is a file that tells git to ignore stuff; it can be certain files types or entire folder or one specific file.
- **WARNING:** The file upload limit for GitHub is **100 Mb**. Do NOT upload data to GitHub. We are using GitHub for documentation purposes only.

2. Make a `.gitignore`

Adding a `.gitignore`

1. What is a `.gitignore`?

- A `.gitignore` is a file that tells git to ignore stuff; it can be certain files types or entire folder or one specific file.
- **WARNING:** The file upload limit for GitHub is **100 Mb**. Do NOT upload data to GitHub. We are using GitHub for documentation purposes only.

2. Make a `.gitignore`

```
1. nano .gitignore
```


Adding a `.gitignore`

1. What is a `.gitignore`?

- A `.gitignore` is a file that tells git to ignore stuff; it can be certain files types or entire folder or one specific file.
- **WARNING:** The file upload limit for GitHub is **100 Mb**. Do NOT upload data to GitHub. We are using GitHub for documentation purposes only.

2. Make a `.gitignore`

1. `nano .gitignore`
2. Add a file type, for example, `.fastq.gz` for it to ignore

Adding a `.gitignore`

1. What is a `.gitignore`?

- A `.gitignore` is a file that tells git to ignore stuff; it can be certain files types or entire folder or one specific file.
- **WARNING:** The file upload limit for GitHub is **100 Mb**. Do NOT upload data to GitHub. We are using GitHub for documentation purposes only.

2. Make a `.gitignore`

1. `nano .gitignore`

2. Add a file type, for example, `.fastq.gz` for it to ignore

3. For more information, check out the great documentation for `.gitignore` <https://git-scm.com/docs/gitignore>

PRACTICE

Practice pulling and
pushing to and from
GitHub

Resources

- Git Book <https://git-scm.com/book/en/v2>
 - Used this as a reference to write this presentation
 - Much, much more here on advanced topics I didn't get into
- Adding SSH keys to your GitHub account
<https://help.github.com/en/github/authenticating-to-github/connecting-to-github-with-ssh>
- .gitignore documentation <https://git-scm.com/docs/gitignore>