Work Together

with git

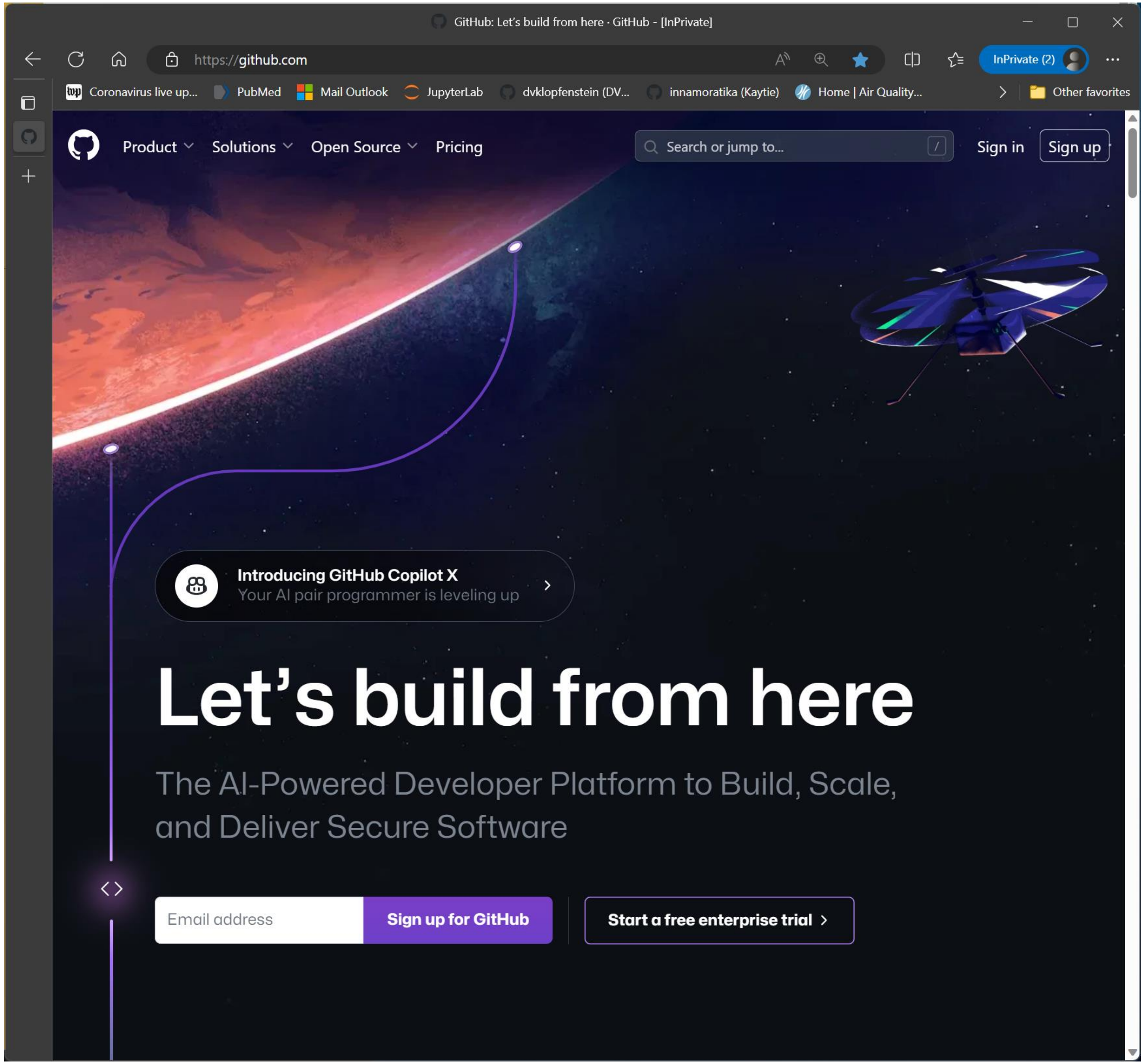
Big Open-Source World
Let us work together

GitHub: Thousands of projects for biology and medicine

- Statistics
- Visualization
- Machine Learning techniques
- Our OMICS class

GitHub

- **100 million users** and growing
- **Protect your work** with version control (git)
- **Work together**
- **Free Open Source projects**:
  - Sequencing
  - Plotting
  - Scientific computing
  - Many more!



Introducing GitHub Copilot X
Your AI pair programmer is leveling up

Let's build from here

The AI-Powered Developer Platform to Build, Scale, and Deliver Secure Software

Collaborate with worldwide researchers

- How many of your colleagues are **NOT LOCAL**?
- How often do you work with local/remote colleagues?
- It is **FUN** to work with people from all over the world
- It is **AFFORDABLE** to work with anyone in the world

How to work with colleagues

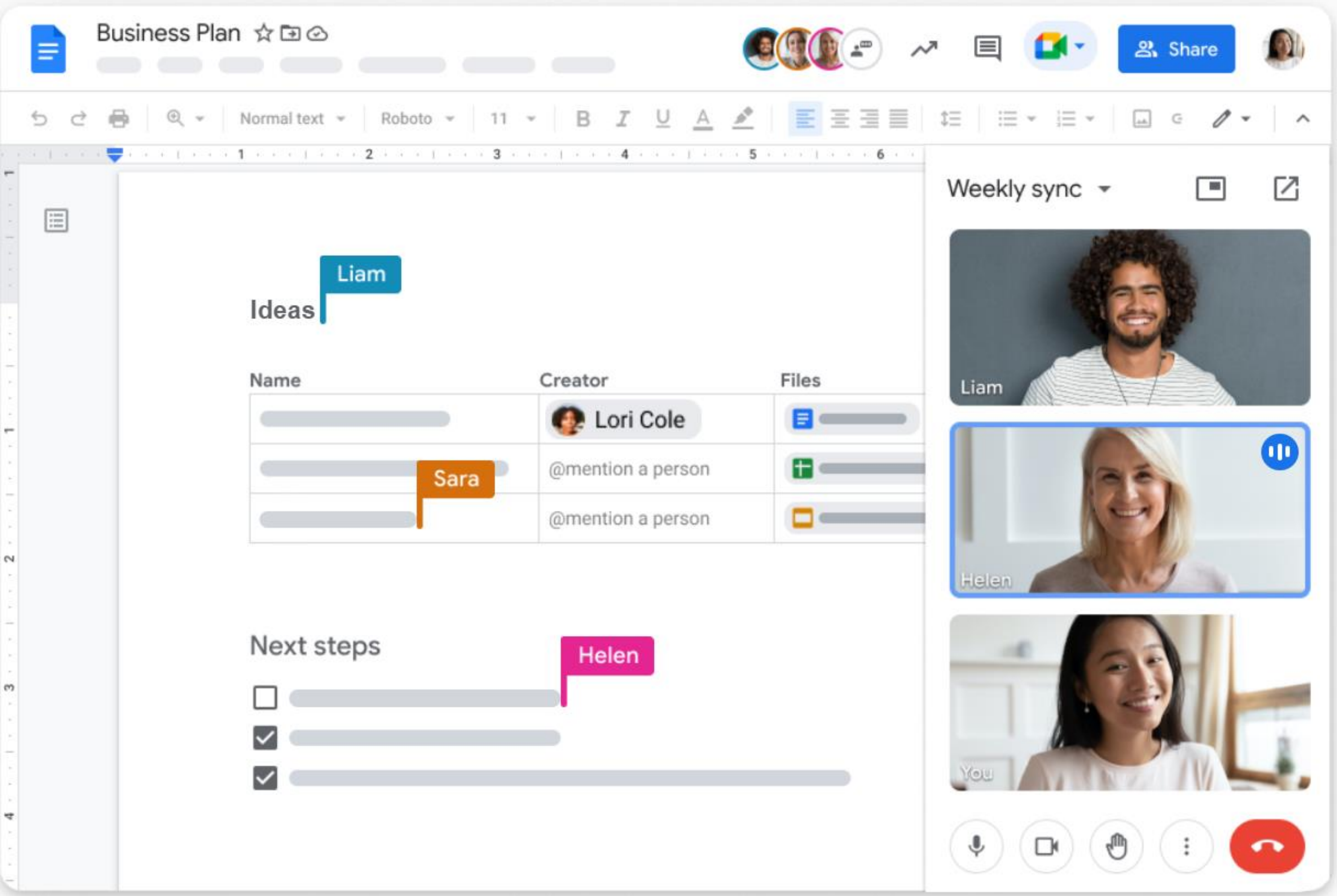
How to work with **remote** and **local** colleagues using version control

Version Control -- Discuss:

- What do you think "version control" is?
- Have you used it before?
- What is a "version" of a file?
- How are versions "controlled"?

Google Docs

- Multiple people can edit
- Tracks versions
- Work together



Ideas

Name	Creator	Files
	Lori Cole	
	@mention a person	
	@mention a person	

Next steps

□
☑
☑

Google Docs vs. GitHub

Google Docs

- Micro versions silently saved automatically
- People see your edits as you work as if they were looking over their shoulder
- People can edit your work while you are working
- You are always sharing your work no matter what

GitHub

- You control when you save versions
- You describe each version
- You work in peace with no one looking over your shoulder while you work
- Only when you are happy with your work, you share it
- Only see other people's edits when convenient for you

What is git version control?

1. Protect your work
   - Recover deleted files
   - Reverse bad edits
   - Track your progress
   - Keep everything in one place
2. Work on projects with colleagues
   - Fun!
   - Use researchers tools they built for their own projects
   - Build better projects together
   - Affordable: Free with a student email; $4/month otherwise

OMICS class repository

https://github.com/DamLabResources/OMICS

GitHub Scavenger Hunt

https://github.com/DamLabResources/OMICS

- Who are the instructors for this course?
- Is there a group project?
- Where is the content for each week?
- What does grading consist of?
- What is in the README file?

Vocabulary of git: Eight words

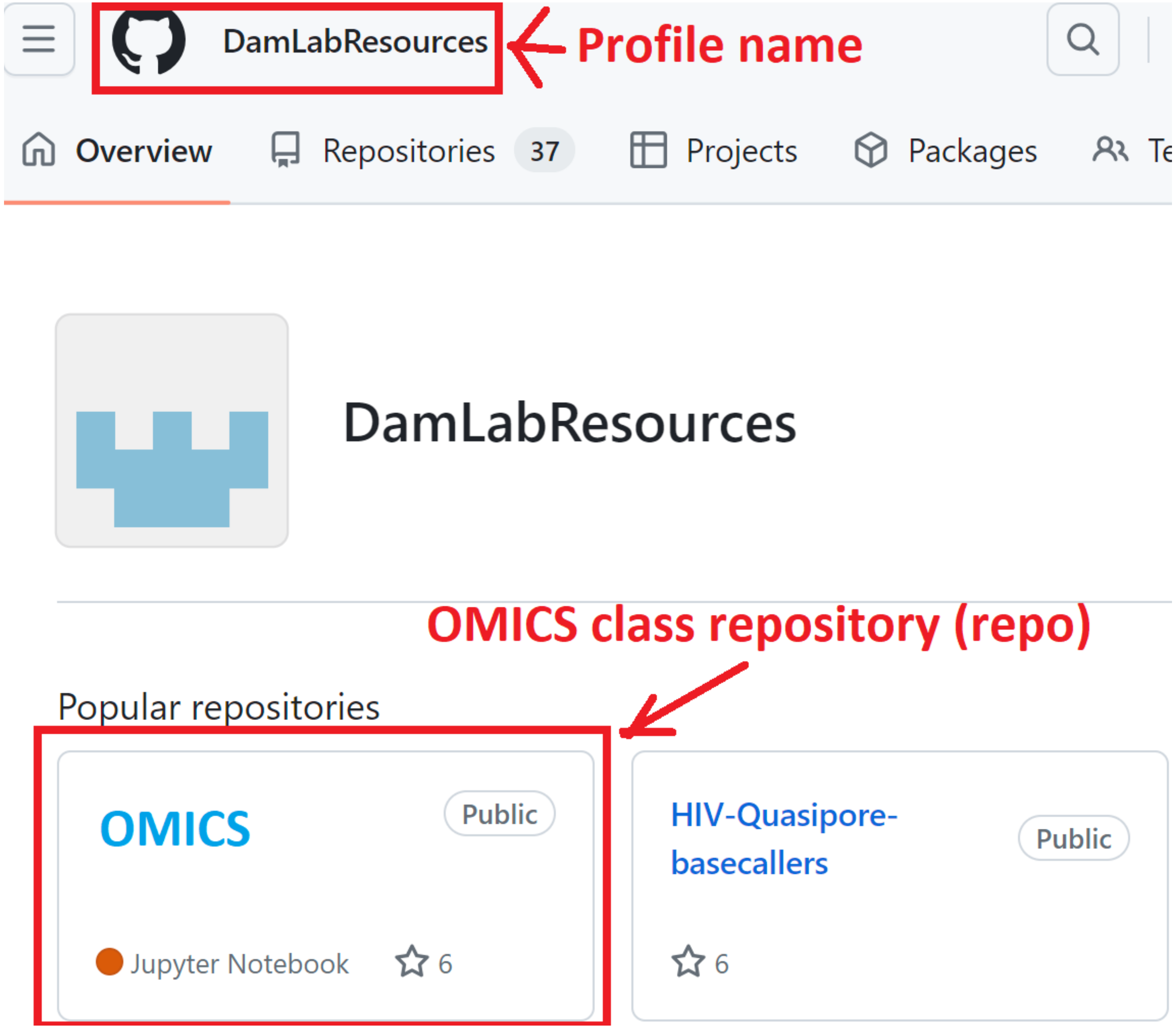
- **Repo**
- **Fork**
- `git` **`clone`**
- `git` **`commit`**
  - `git` **`status`**
  - `git` **`diff`**
- `git` **`push`**
- **Pull Request** (**PR**): Very Prestigious. Resume worthy

Repository (repo)

- Repository:
  - Place, room, or container where something is deposited or stored (Merriam-Webster)
  - A repository contains all your project's files and each file's revision history (docs.github.com)

- "Repo" is short for "repository"

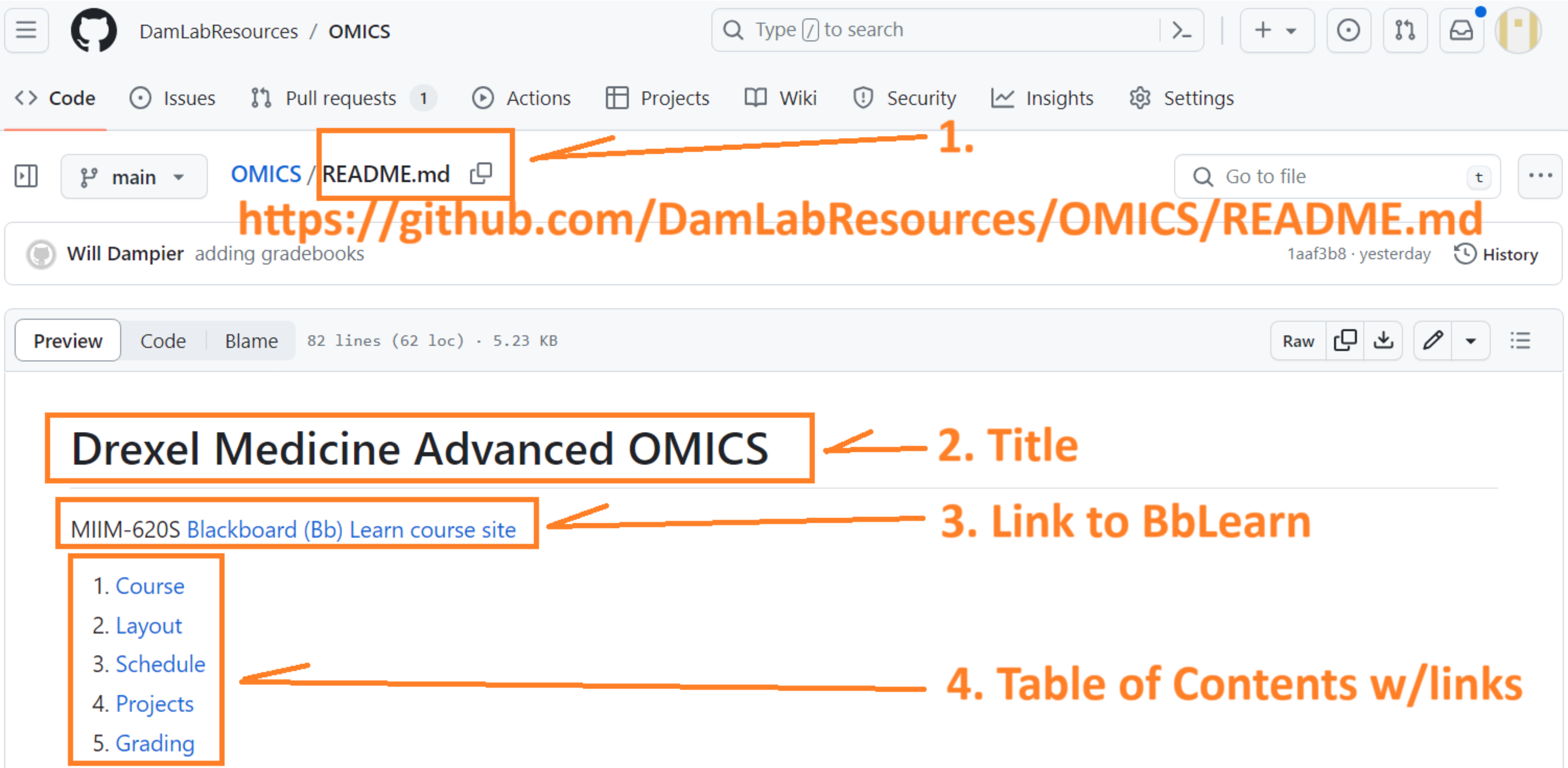
https://github.com/DamLabResources/OMICS



Profile name

OMICS class repository (repo)

Repo README.md



1. https://github.com/DamLabResources/OMICS/README.md

Drexel Medicine Advanced OMICS

2. Title

MIIM-620S Blackboard (Bb) Learn course site

3. Link to BbLearn

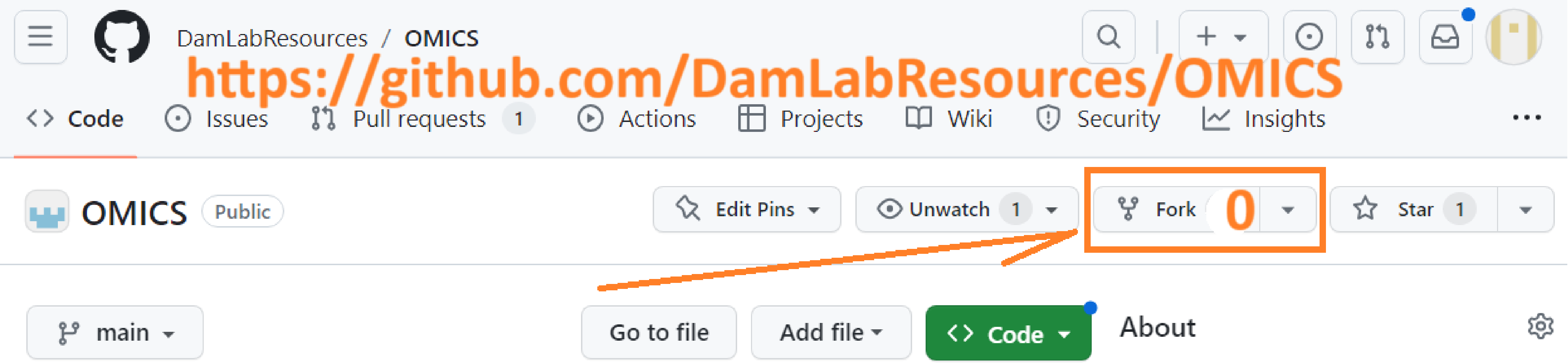
1. Course
2. Layout
3. Schedule
4. Projects
5. Grading

4. Table of Contents w/links

Vocabulary of git: Eight words

- **Repo**
- **Fork**
- `git` **`clone`**
- `git` **`commit`**
  - `git` **`status`**
  - `git` **`diff`**
- `git` **`push`**
- **Pull Request** (**PR**): Very Prestigious. Resume worthy

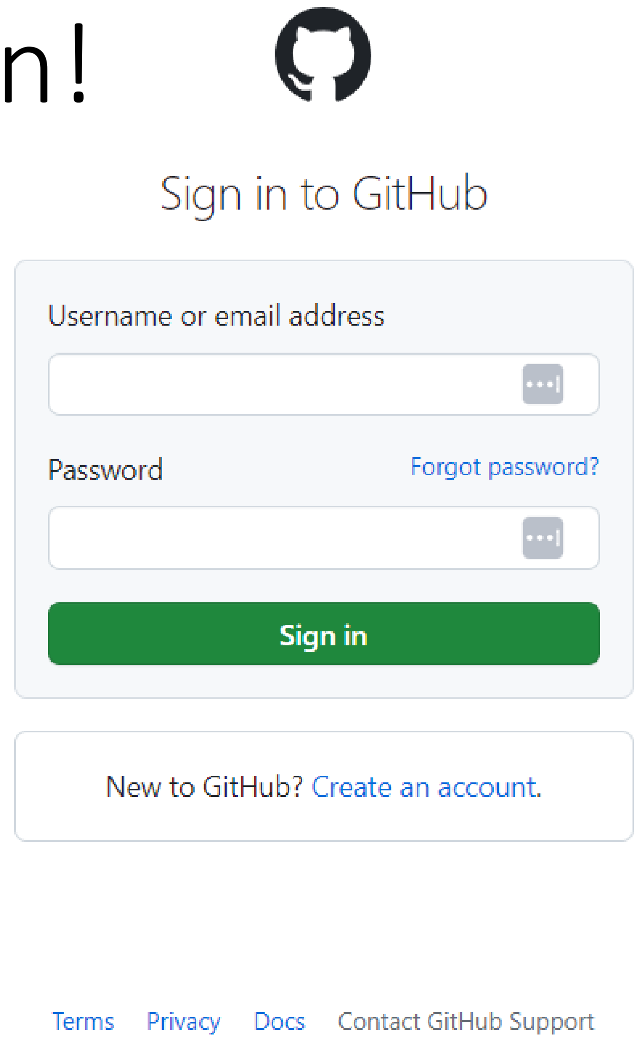
What happens when you click on "fork?"



https://github.com/DamLabResources/OMICS

Time to make a GitHub login!

- Best practices for choosing a Git name:
  1. Professional
     - First initials plus last name
  2. Fun!
     - Easy to remember
     - Individual to you
     - Techy or gamer style



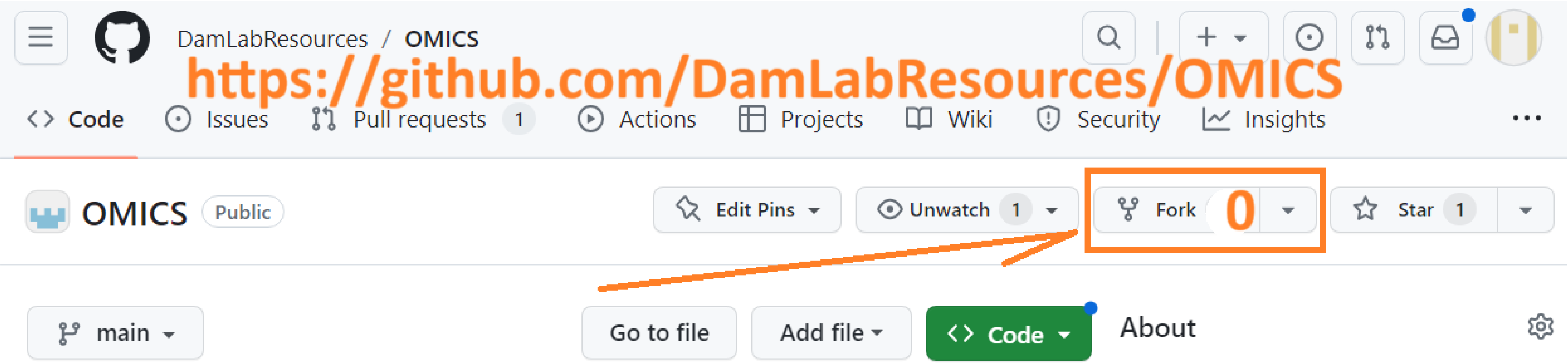
Sign in to GitHub

Username or email address

Password

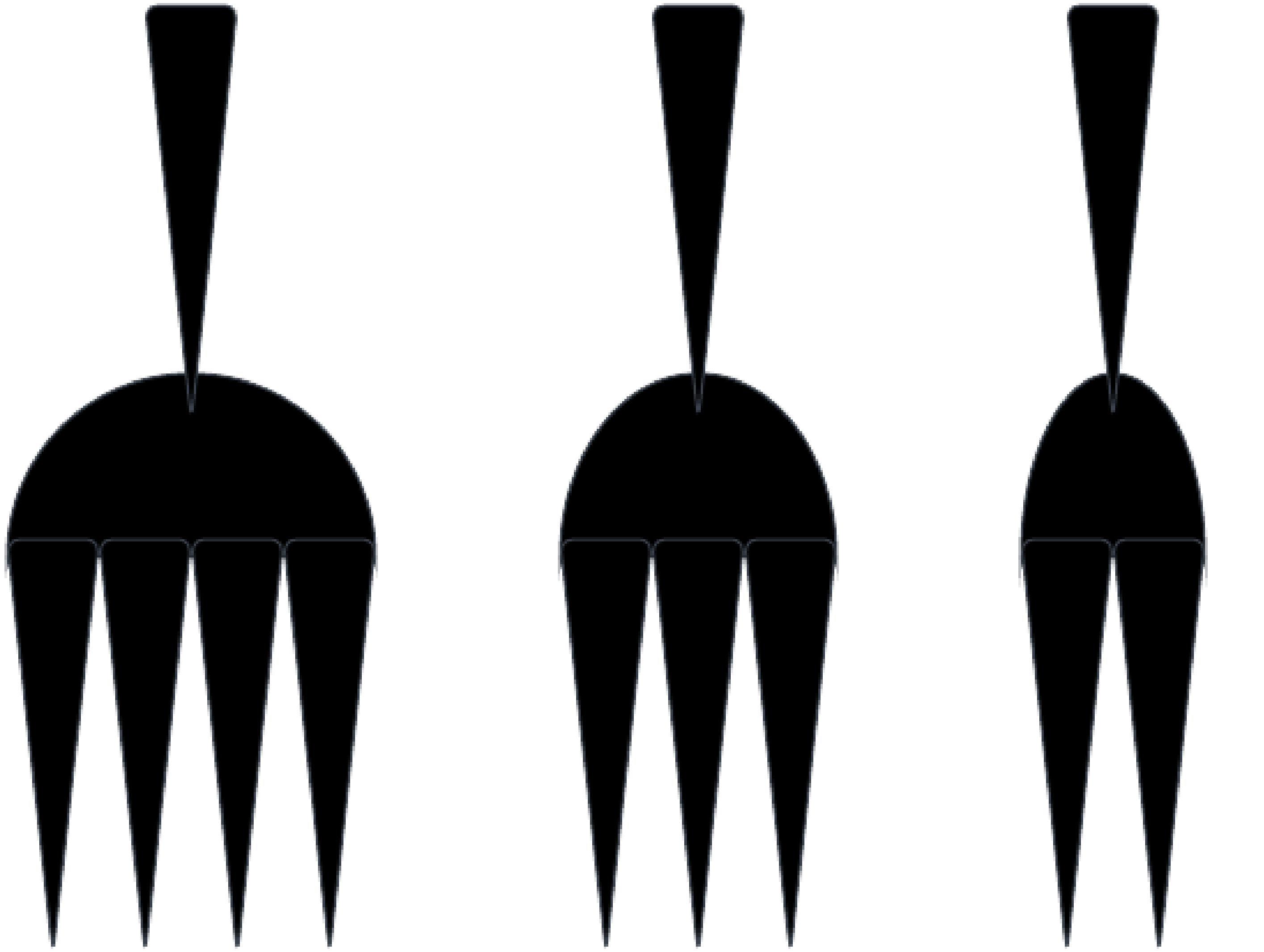
New to GitHub? Create an account.

Let's try this again- NOW what happens when you click "fork"?

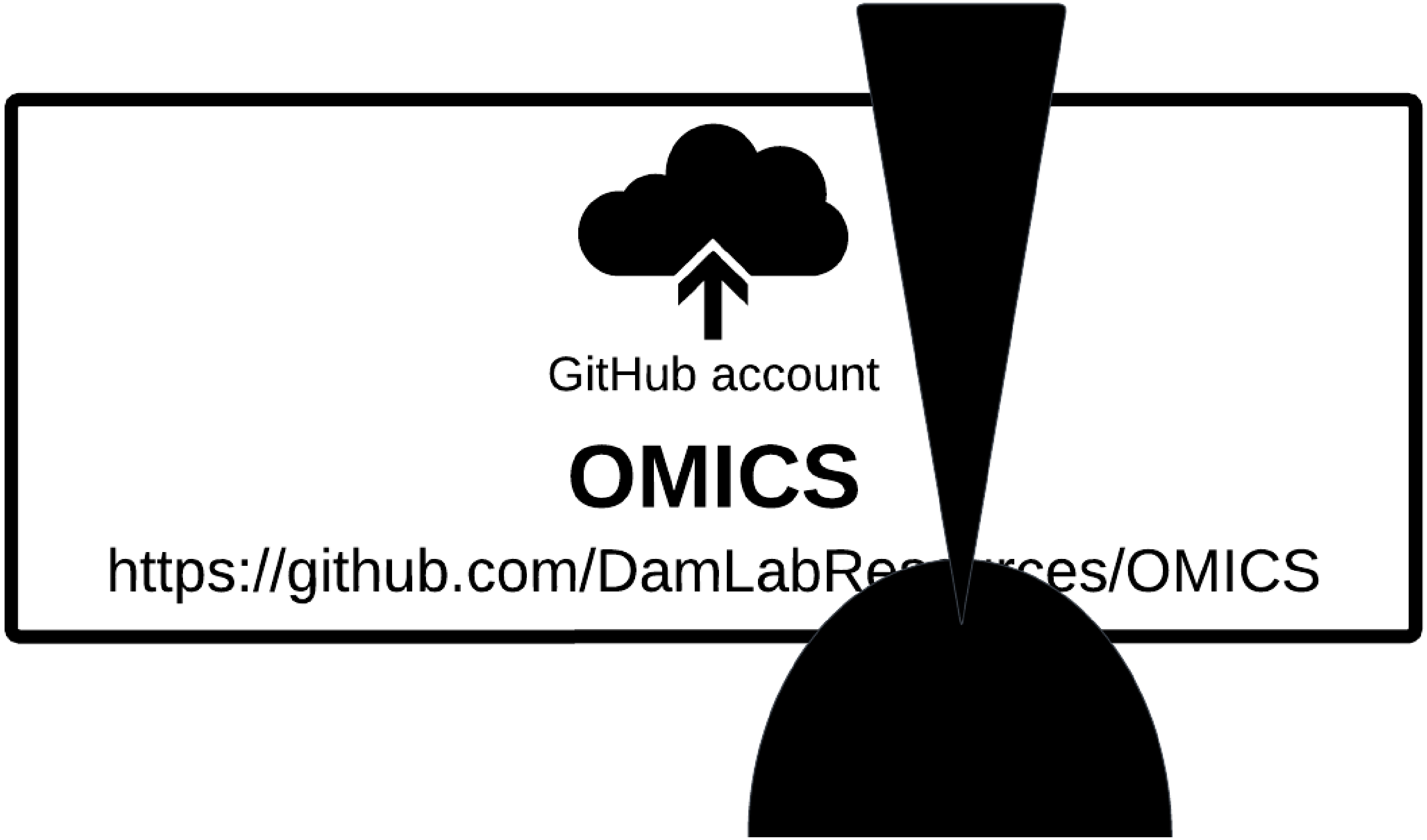


https://github.com/DamLabResources/OMICS

Fork



Zero Forks = 0 tines on the fork



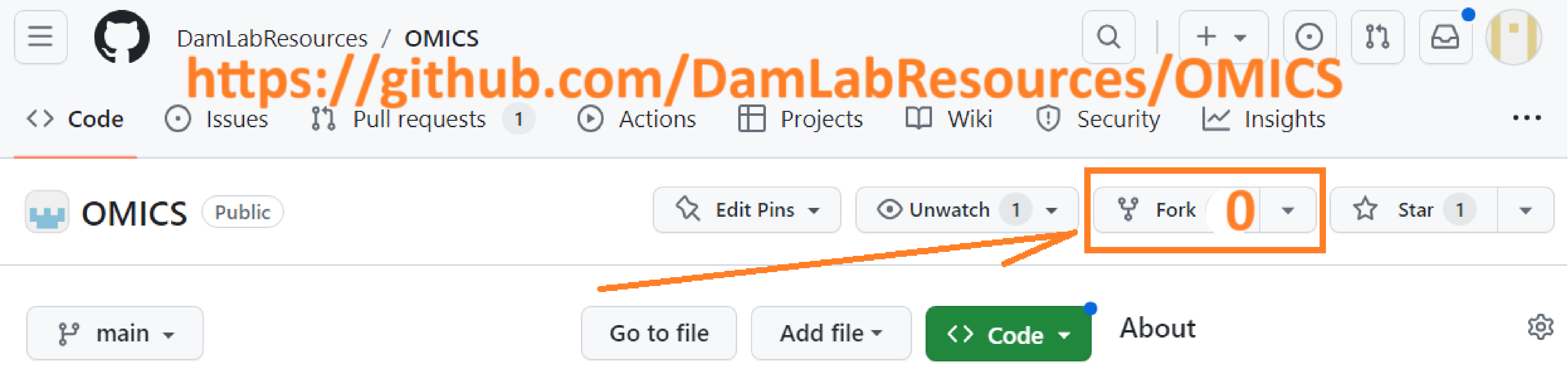
GitHub account

OMICS

https://github.com/DamLabRes rces/OMICS

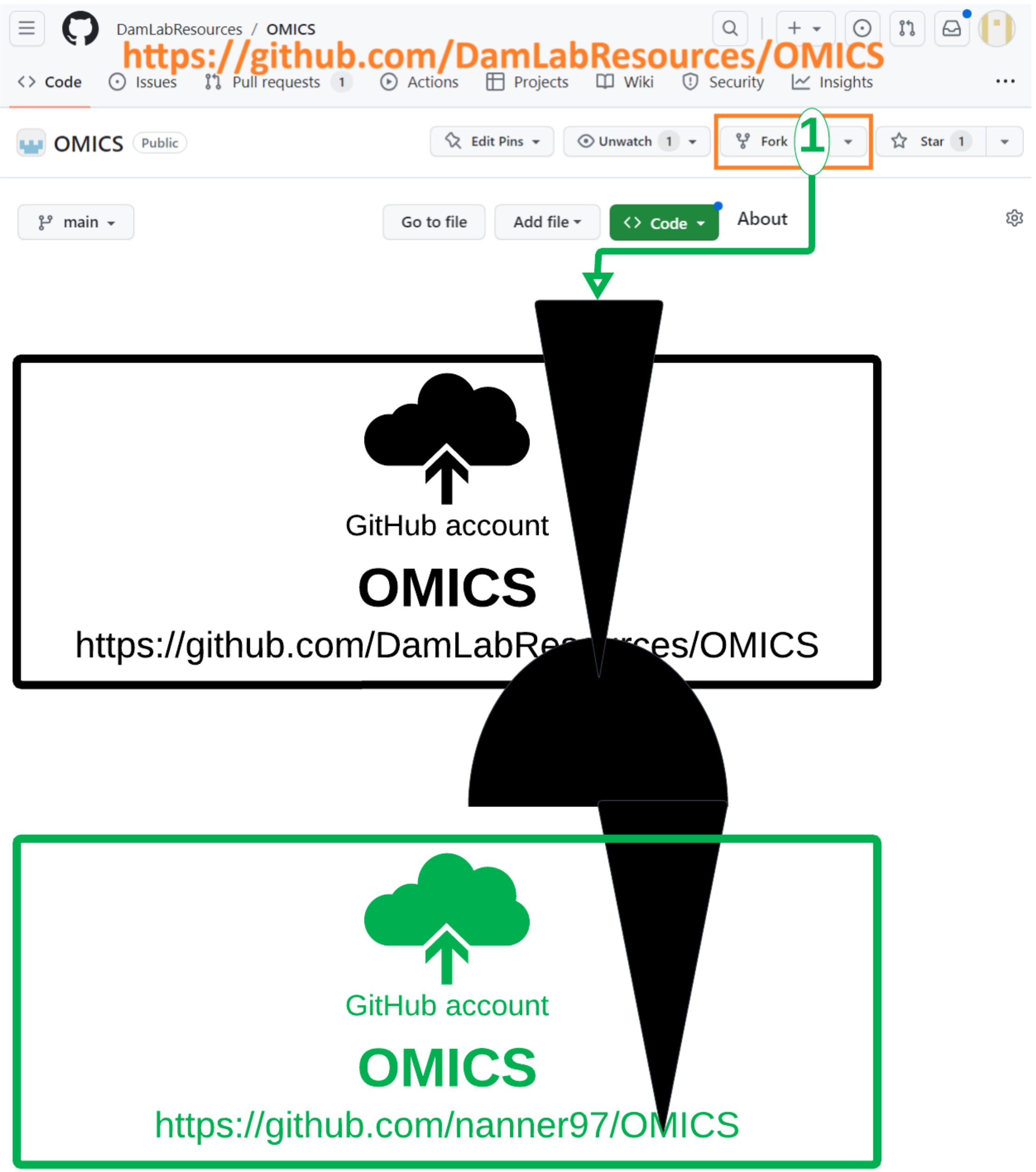
Make a fork (copy)

Make a GitHub account (or sign in if you have one already) and click the "fork" button on the OMICS repo.
What happens?



https://github.com/DamLabResources/OMICS

Fork



https://github.com/DamLabResources/OMICS

GitHub account

OMICS

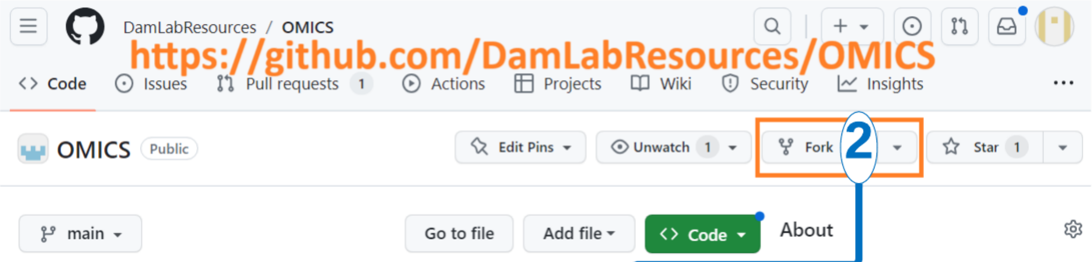
https://github.com/DamLabResources/OMICS

GitHub account

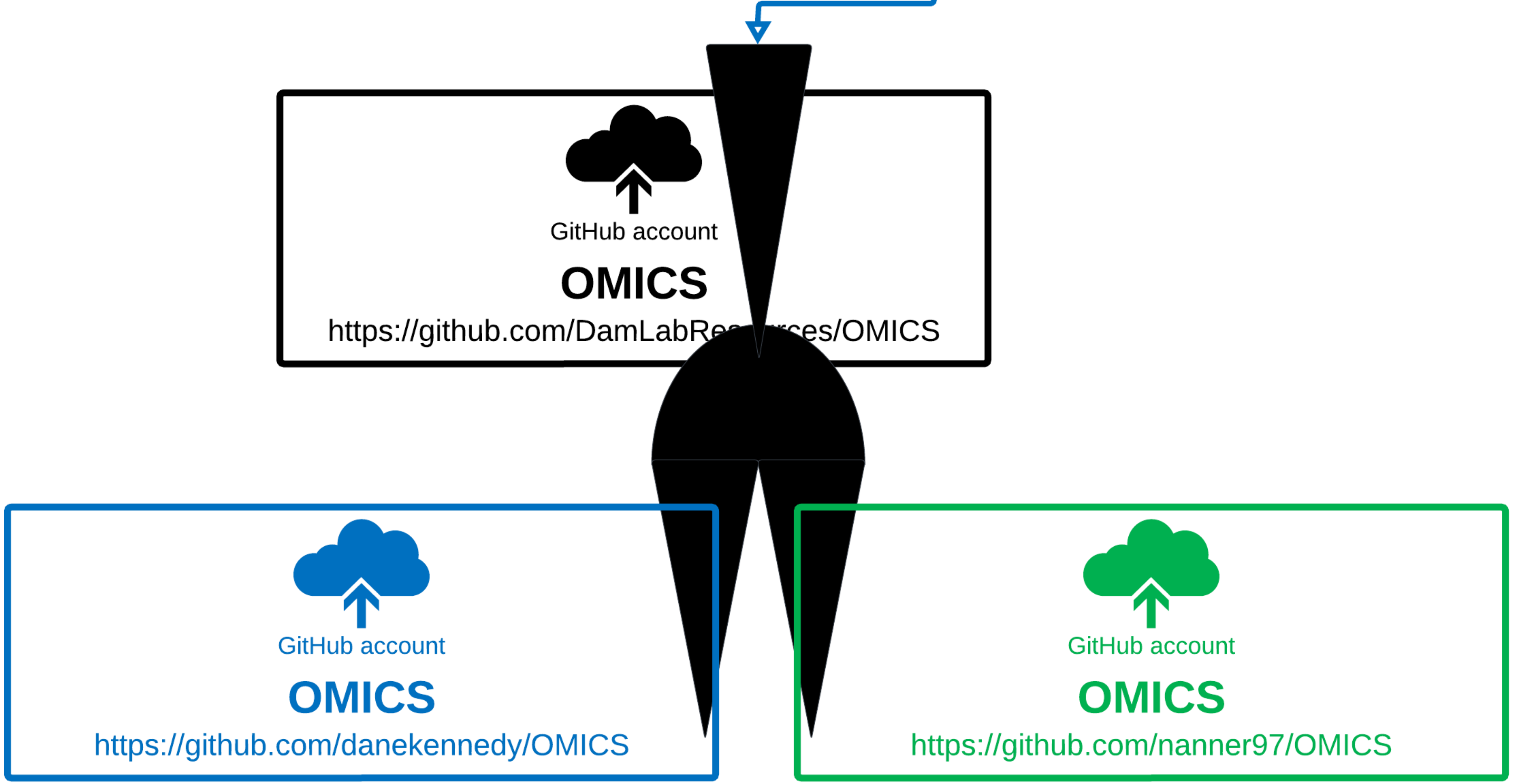
OMICS

https://github.com/nanner97/OMICS

Fork



https://github.com/DamLabResources/OMICS



GitHub account

OMICS

https://github.com/DamLabResources/OMICS

GitHub account

OMICS

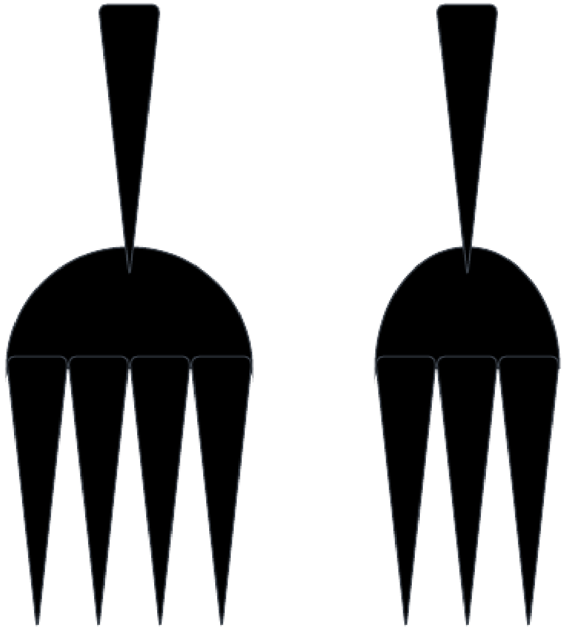
https://github.com/danekennedy/OMICS

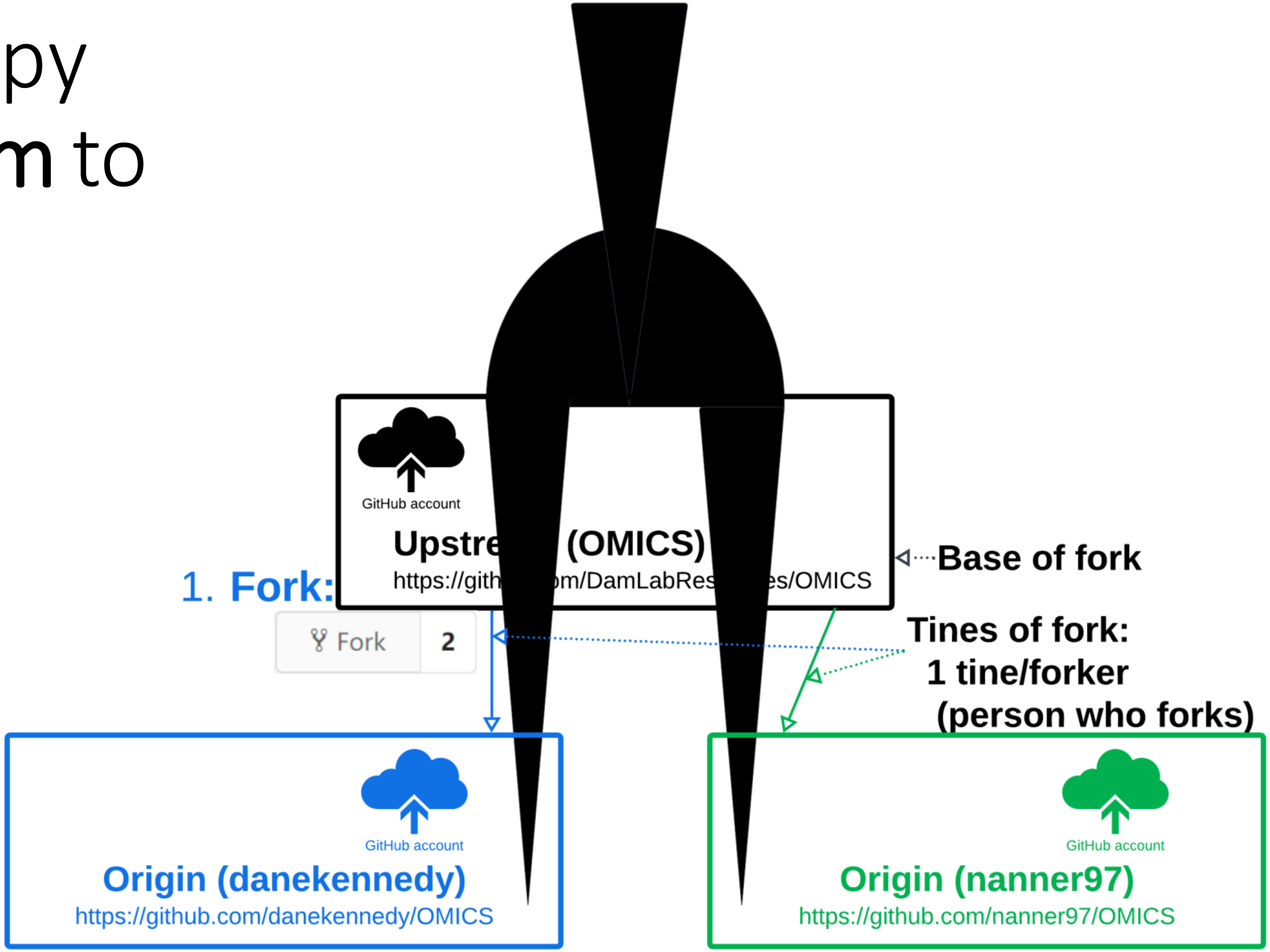
GitHub account

OMICS

https://github.com/nanner97/OMICS

1. Fork: Copy **Upstream** to **Origin**





GitHub account

Upstre (OMICS)

https://gith om/DamLabRes es/OMICS

Base of fork

1. Fork:

Fork 2

Tines of fork:
1 tine/forker
(person who forks)

GitHub account

Origin (danekennedy)

https://github.com/danekennedy/OMICS

GitHub account

Origin (nanner97)

https://github.com/nanner97/OMICS

Vocabulary of git: Eight words

- **Repo** **# Project, Repository storing project**
- **Fork** **# Copy**
- `git` **`clone`**
- `git` **`commit`**
  - `git` **`status`**
  - `git` **`diff`**
- `git` **`push`**
- **Pull Request** (**PR**): Very Prestigious. Resume worthy

Lab Activity 1: Log in to JupyterHub

JupyterHub: Computing on the web!

- Coding doesn't happen on GitHub- it happens elsewhere, and gets uploaded to GitHub
- In this case, we use a web-based program called JupyterHub

Lab activity 1:

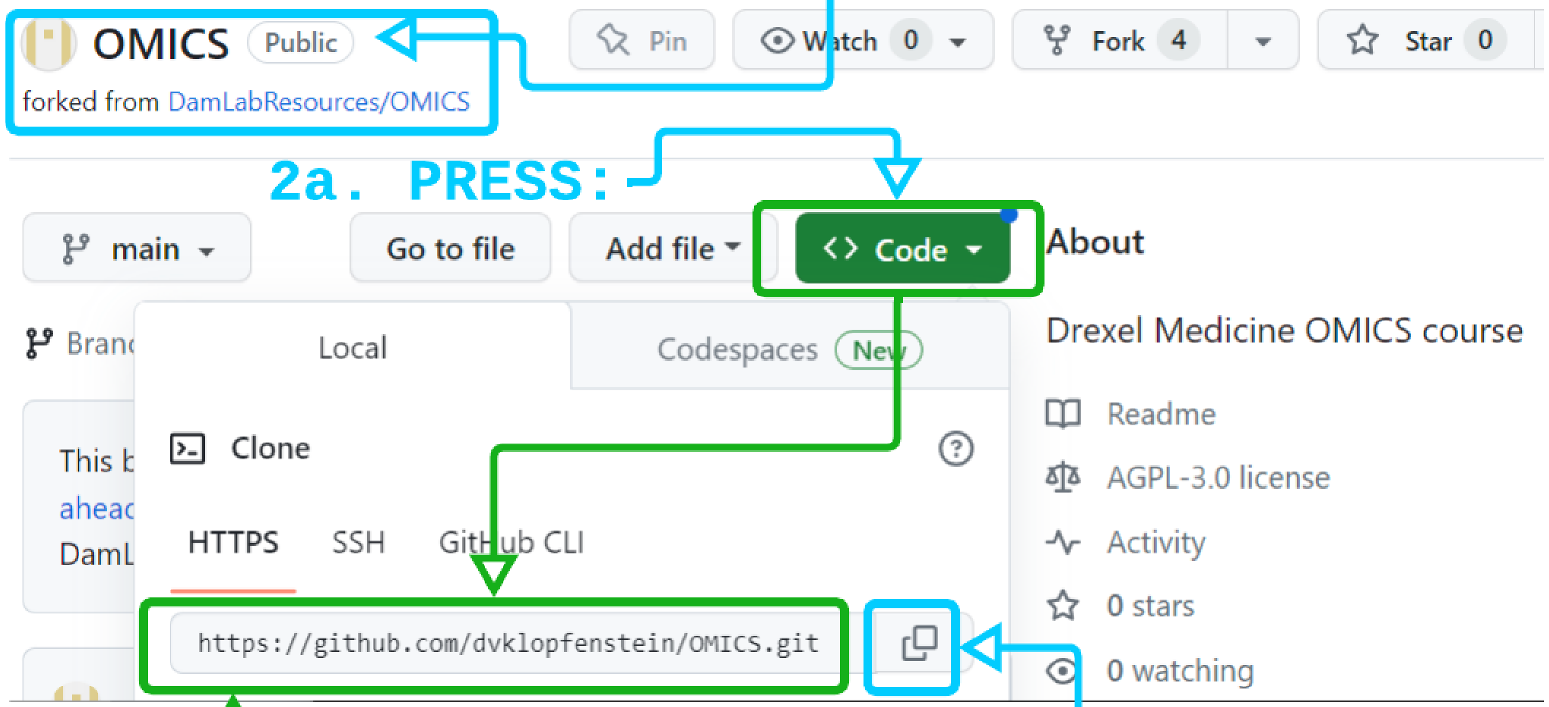
1. **Get VPNed (**https://vpn.drexel.edu/) **into the network (drexelmed.edu)**
2. **Create an account into the MistakeNot server:** http://10.11.19.48/
    1. JupyterHub- username and password (use your git login name)
3. **Now- welcome to Linux Land:** File->New->Terminal

Vocabulary of git: Eight words

- **Repo: Upstream, Origin, & Local**
- **Fork:** # Copy **Upstream -> Origin**
- `git` **`clone`** `[URL]` # Copy **Origin -> Local**
- `git` **`commit`**
  - `git` **`status`**
  - `git` **`diff`**
- `git` **`push`**
- **Pull Request** (**PR**): Very Prestigious. Resume worthy

```
$ cd ~/repos
$ git clone [URL]
```

2. CLONE from ORIGIN:

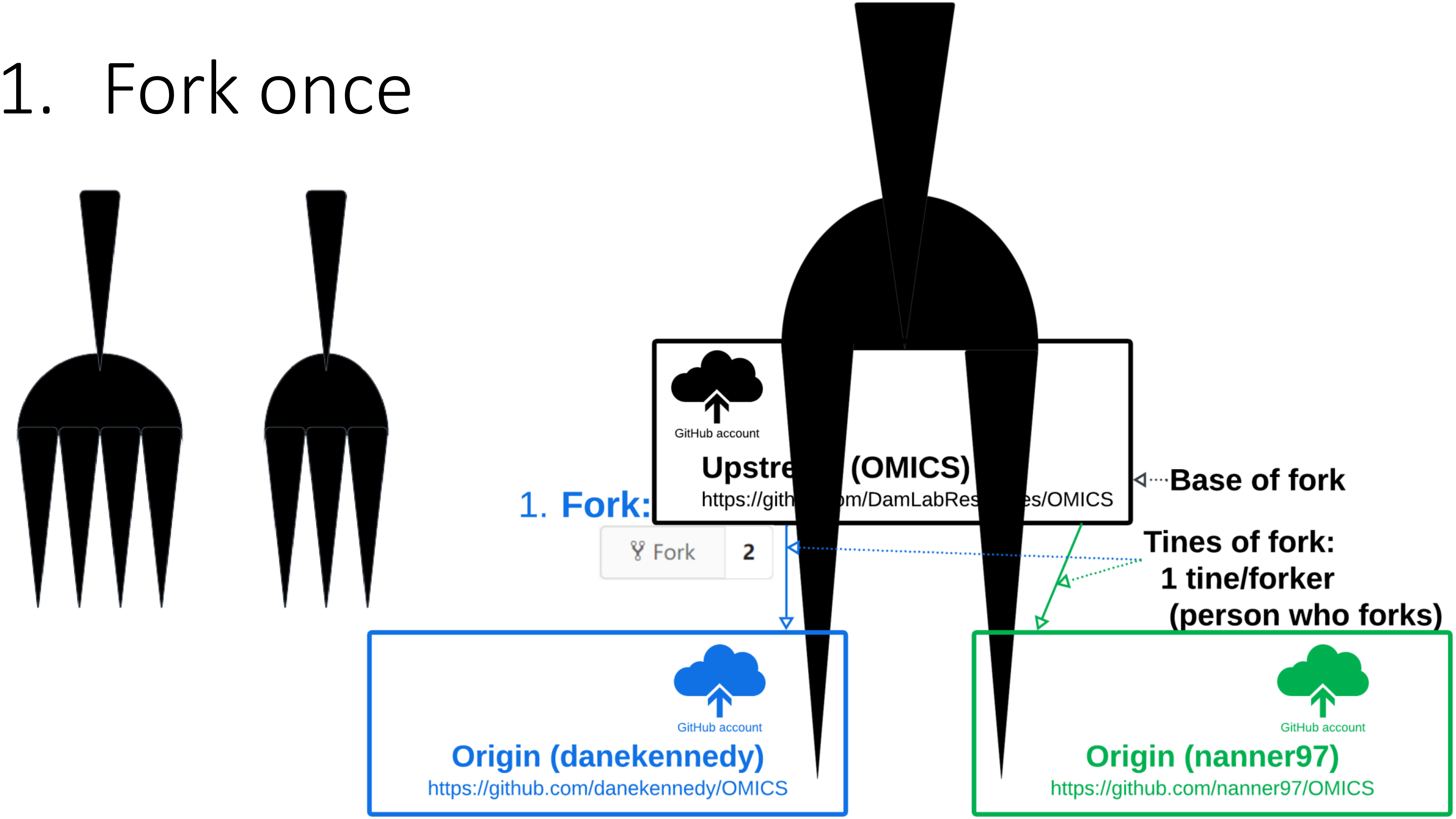


2a. PRESS:

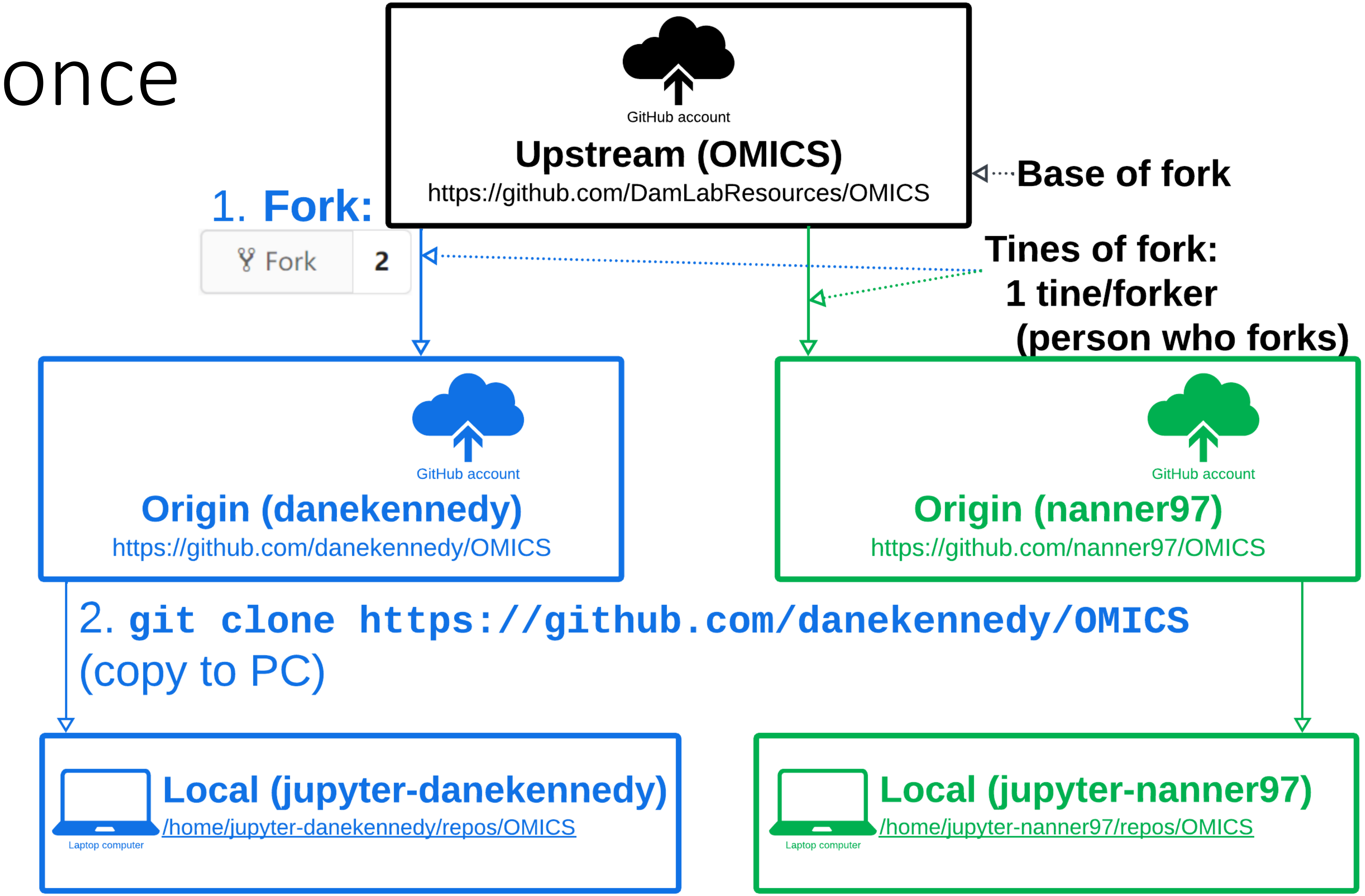
2b. COPY URL; PRESS:

```
https://github.com/yourgitname/OMICS.git
```

1. Fork once


GitHub account
Upstre (OMICS)
https://gith om/DamLabRes es/OMICS
Base of fork
1. Fork:
Fork 2
Tines of fork:
1 tine/forker
(person who forks)
GitHub account
Origin (danekennedy)
https://github.com/danekennedy/OMICS
GitHub account
Origin (nanner97)
https://github.com/nanner97/OMICS

2. Clone once



1. **Fork:**

Upstream (OMICS)

https://github.com/DamLabResources/OMICS

Base of fork

Tines of fork:
1 tine/forker
(person who forks)

Origin (danekennedy)

https://github.com/danekennedy/OMICS

Origin (nanner97)

https://github.com/nanner97/OMICS

2. `git clone https://github.com/danekennedy/OMICS`
(copy to PC)

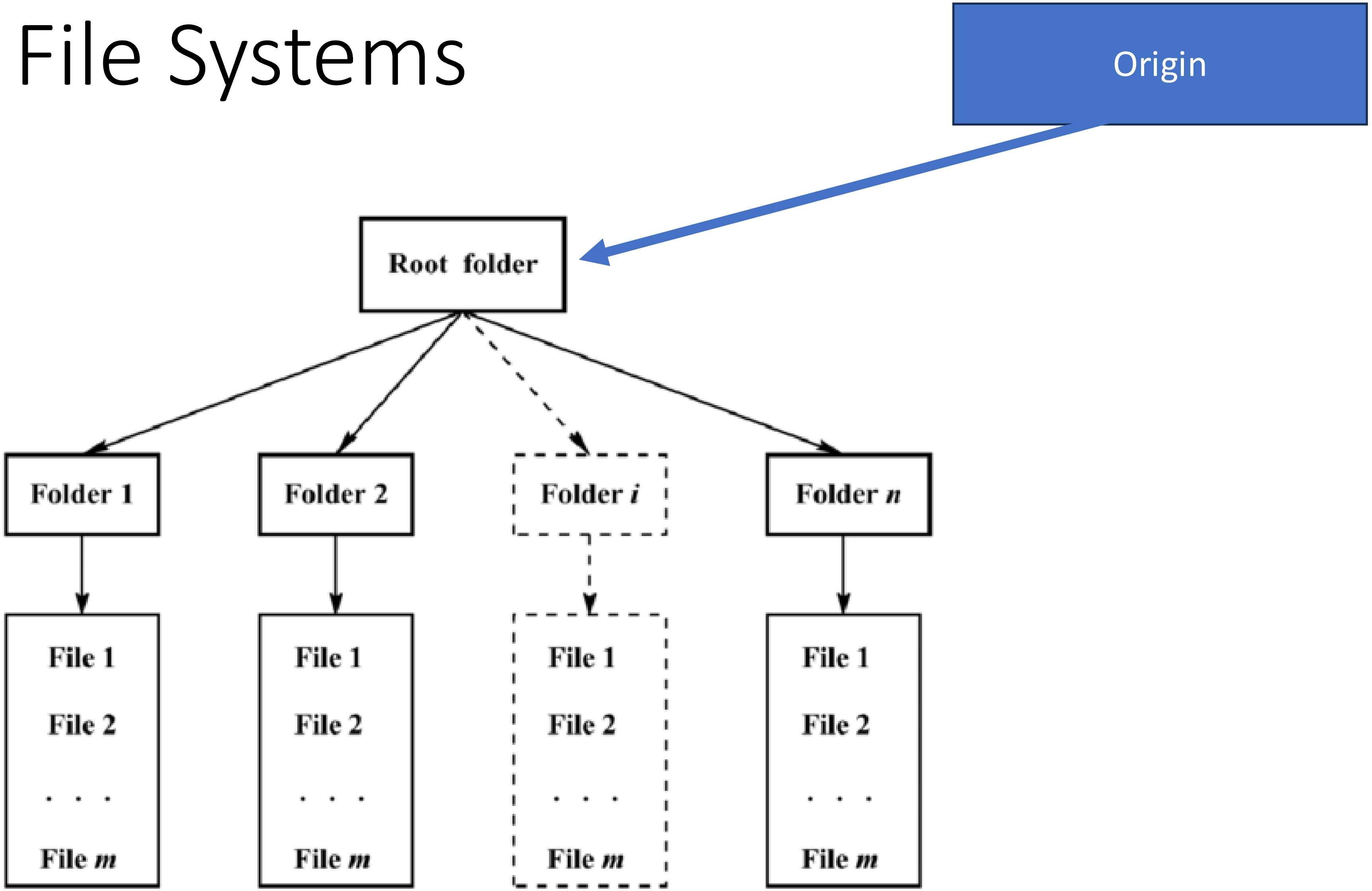
Local (jupyter-danekennedy)

/home/jupyter-danekennedy/repos/OMICS

Local (jupyter-nanner97)

/home/jupyter-nanner97/repos/OMICS

Linux and File Systems



Origin

https://www.researchgate.net/publication/350296947_Assessment_of_Construction_Materials_Trends_Salt_Content_from_Scanned_Images

Lab activity 2: Welcome to the Command Line

- `pwd`
- `cd ~/repos/OMICS`
- `ls`
- `ls –help`
- `ls –lrt`
- `ls -lrth`
- `cd`
- `cd repo`
- `cd ~`
- `cd ~/repo/OMICS/projects`

Lab activity 2: Welcome to the Command Line

- Create: `README_[GitHubLogin].md`
- What is a README?
  - Welcome!
  - We have been expecting you
  - Let me show you around
- `$ nano README_[GitHubLogin].md`

Lab Activity 3: "History"

- `$ history`
  - What happens when you type this command?
- Take the output of history and save it in your README
  - `README_[GitHubLogin].md`

- New commands: history, less

"Peeking" at a file

- `$ less README_[GitHubLogin].md`
- `$ more README_[GitHubLogin].md`
- `$ head README_[GitHubLogin].md`
- `$ tail README_[GitHubLogin].md`
- `$ echo README_[GitHubLogin].md`
- `$ cat README_[GitHubLogin].md`

What do each of these commands do?

Vocabulary of git: Eight words

- **Repo: Upstream, Origin, & Local**
- **Fork:** # Copy **Upstream -> Origin**
- `git clone [URL]` # Copy **Origin -> Local**
- `git commit`
    - `git status`
    - `git diff`
- `git push`
- **Pull Request** (**PR**): Very Prestigious. Resume worthy

Vocabulary of git: Eight words

- **Repo: Upstream, Origin, & Local**
- **Fork: Upstream -> Origin**
- `git clone [URL]  # Origin -> Local`
- `git commit`
  - `git add`
  - `git status`
  - `git diff`
- `git push`
- **Pull Request (PR)**: Very Prestigious. Resume worthy

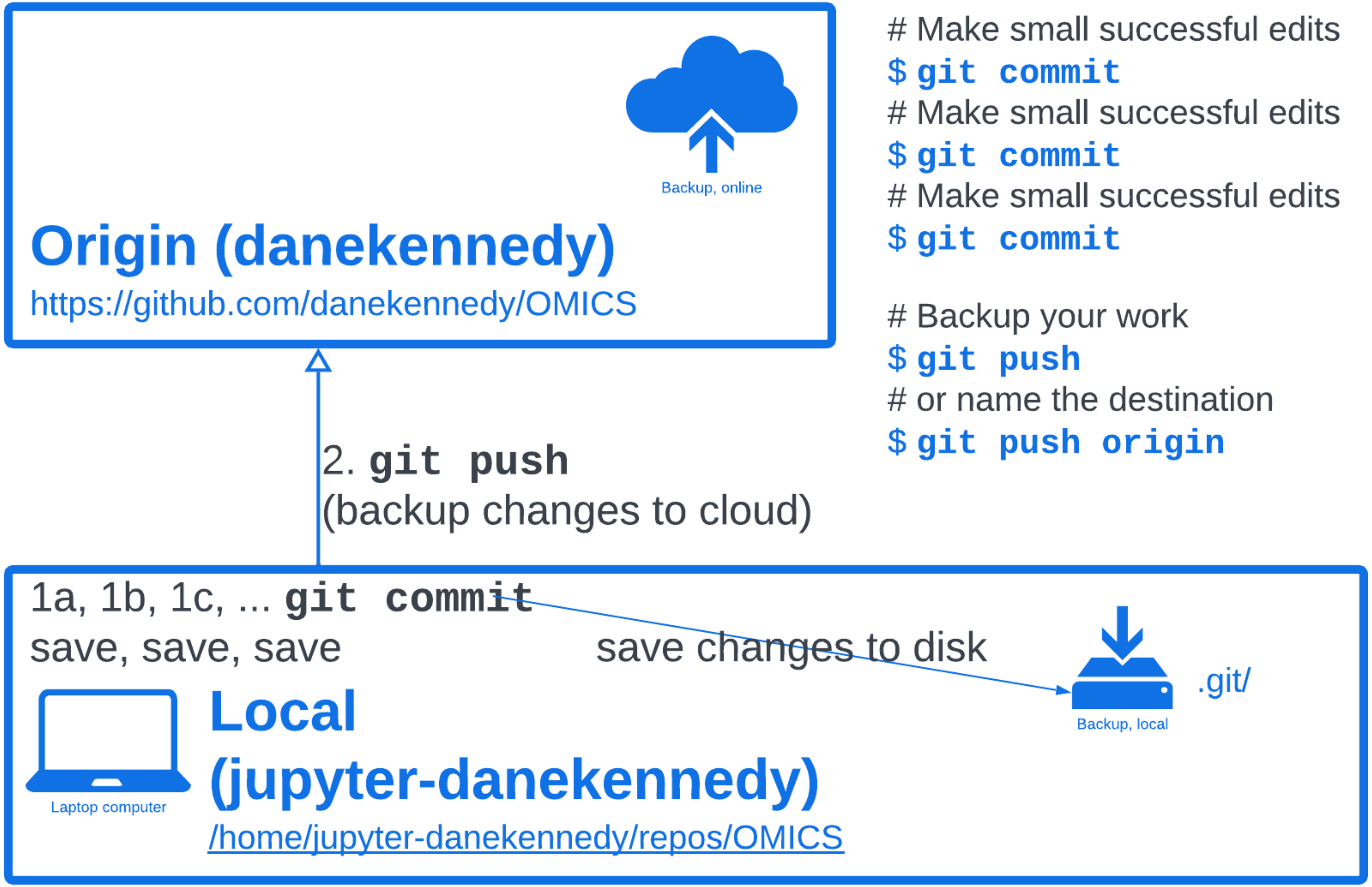
JupyterHub- commit your README

- Where is it? What happened when you committed it?

Vocabulary of git: Eight words

- **Repo: Upstream, Origin, & Local**
- **Fork: Upstream -> Origin**
- `git clone [URL]  # Origin -> Local`
- `git commit #`
  - `git add README_mygitlogin.md   #`
  - `git status #`
  - `git diff #`
- `git push`
- **Pull Request** (**PR**): Very Prestigious. Resume worthy

Daily Flow



```
# Make small successful edits
$ git commit
# Make small successful edits
$ git commit
# Make small successful edits
$ git commit

# Backup your work
$ git push
# or name the destination
$ git push origin
```

Git push- where are my files NOW?

Go to origin and check out if anything has changed!

Git add, commit, push

- `$ cd ~/repos/OMICS/project/[mygitlogin]`
- Make a new file: README.md
- `$ git add README.md`
- Edit README.md
- `$ git commit !$ --message 'My README'`
- `$ git commit README.md -m 'My README'`
- `$ git commit –help`
- `$ git push`
- Go look: https://github.com/danekennedy/OMICS/doc

Vocabulary of git: Eight words

- **Repo: Upstream, Origin, & Local**
- **Fork: Upstream -> Origin**
- `git clone [URL]   # Origin -> Local`
- `git commit        # Local -> repo_name/.git`
  - `git status      # New, Modified, Untracked`
  - `git diff        # Show lines Added/Deleted`
- `git push          # repo_name/.git -> Origin`
- **Pull Request (PR)**: Very Prestigious. (next week!)

Your project README.md

Add it here:

OMICS/projects/[yourgithubname]/README.md

Your Independent Project

- Analyze genomic data week-by-week
- Week 1:
  - Understand the file structure of the OMICS course project stored in GitHub
  - Download your writable copy of the OMIC course project
  - Add your sub-directory to OMICS/projects:
    - OMICS/projects/[yourgithubname]
  - Create:
    - OMICS/projects/[yourgithubname]/README.md
  - Save revisions to your README.md as you develop it
  - Back-up the work done on your computer to your GitHub account

The End

Homework: Work on Independent Project Week 1

Links

- https://blog.hubspot.com/website/what-is-github-used-for

Git config: Set your name and email

```
$ git config --global user.name "Your Name Comes Here"
$ git config --global user.email you@yourdomain.example.com
```

For example, to set "NO EMAIL"

```
$ git config --global user.name "DANE Kennedy"
$ git config --global user.email danekennedy@users.noreply.github.com
$ git config --list
```