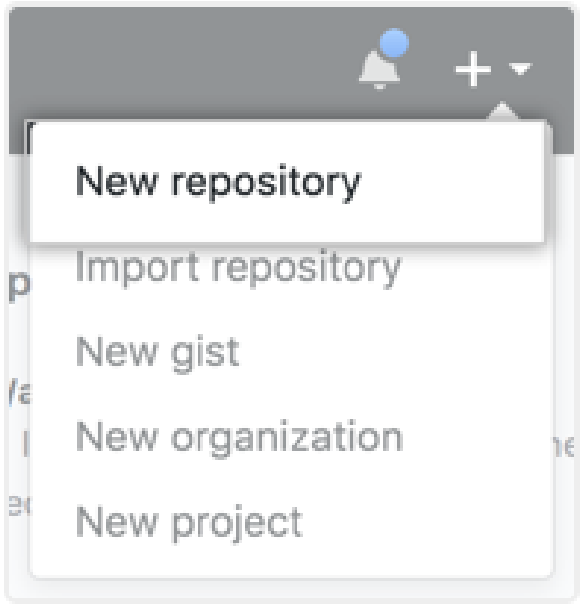
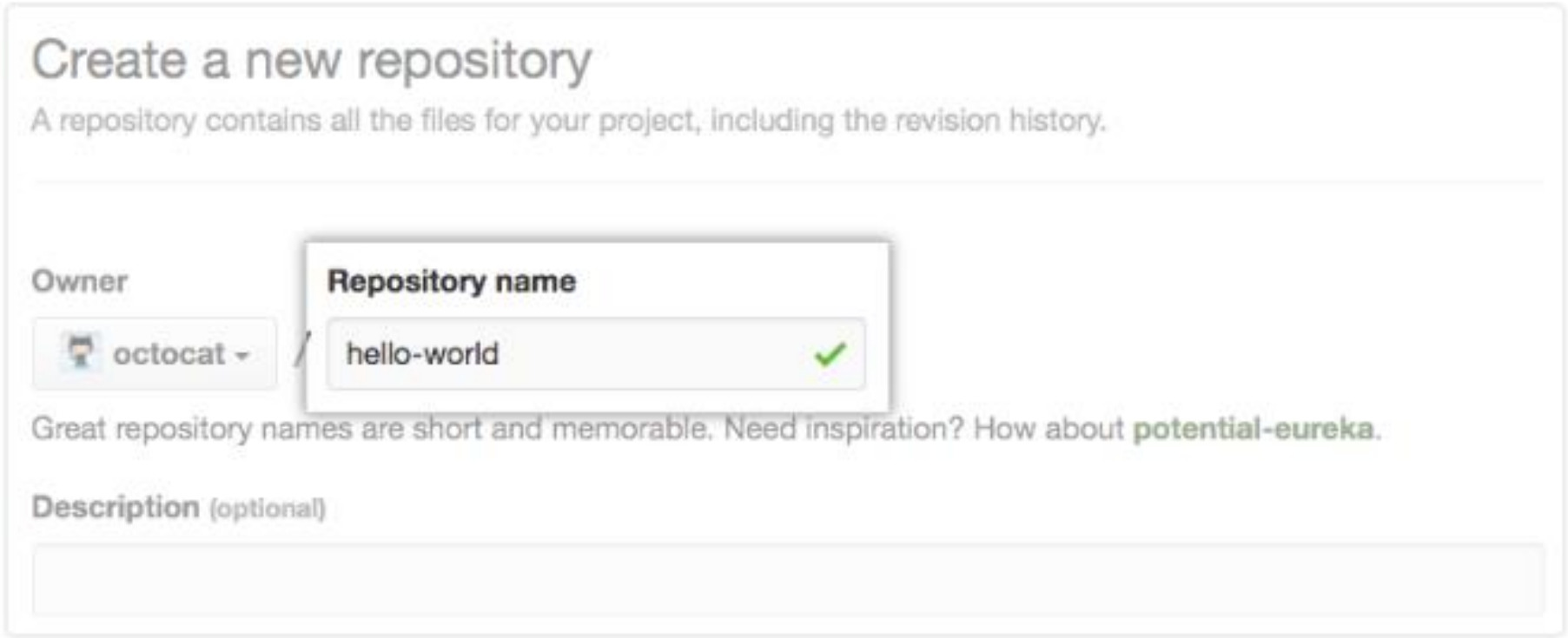
GitHub And Gitkraken

Create a repo

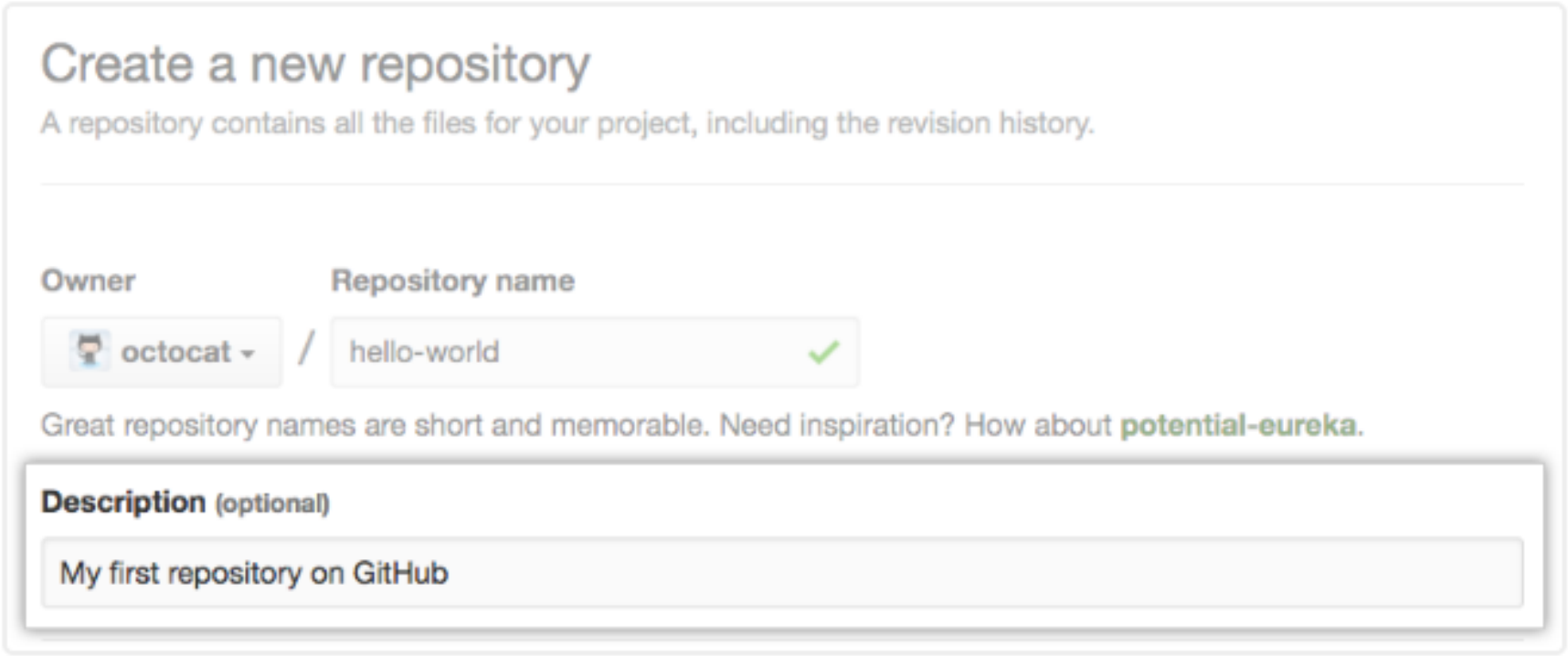
1 In the upper-right corner of any page, click **+**, and then click **New repository**.



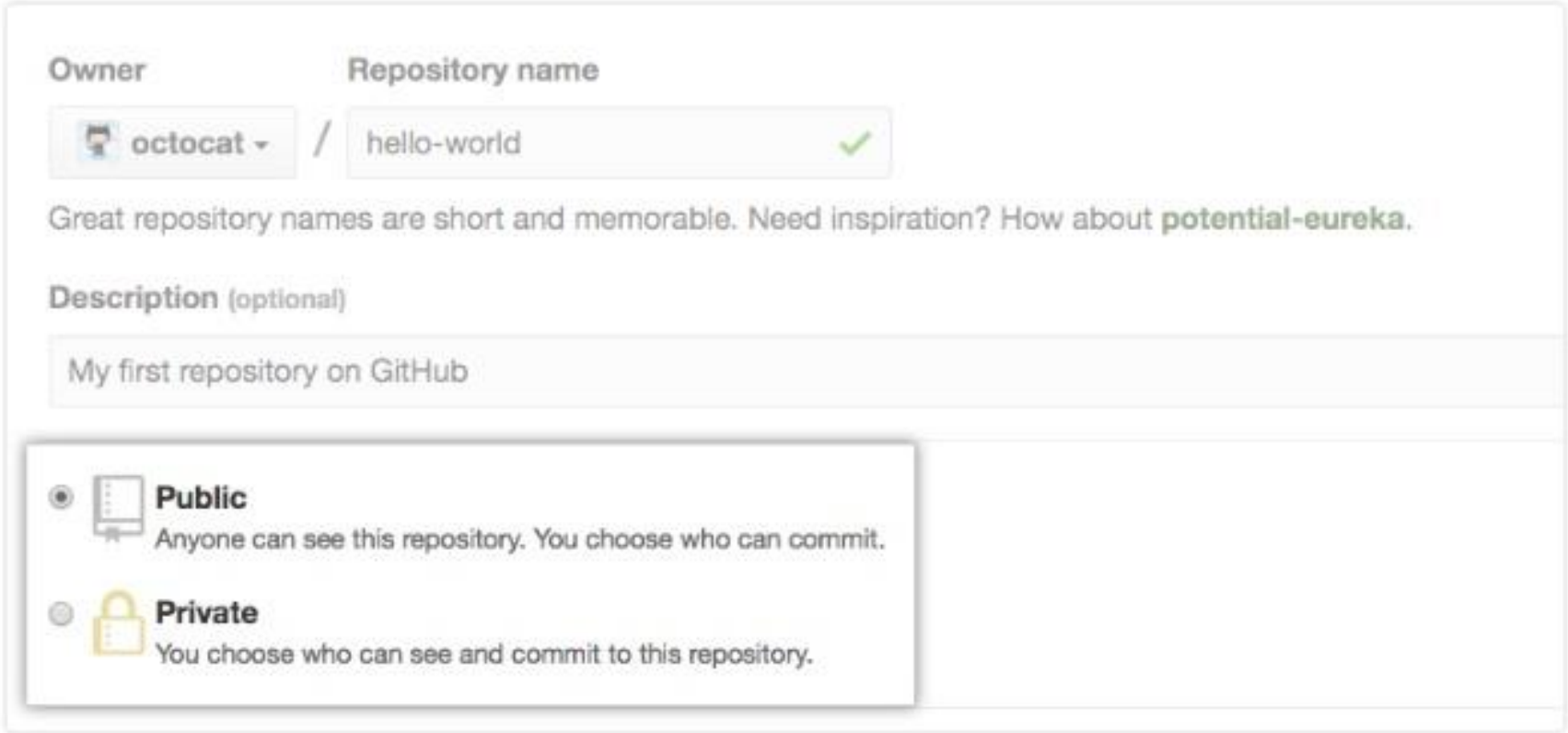
2 Type a short, memorable name for your repository. For example, "hello-world".



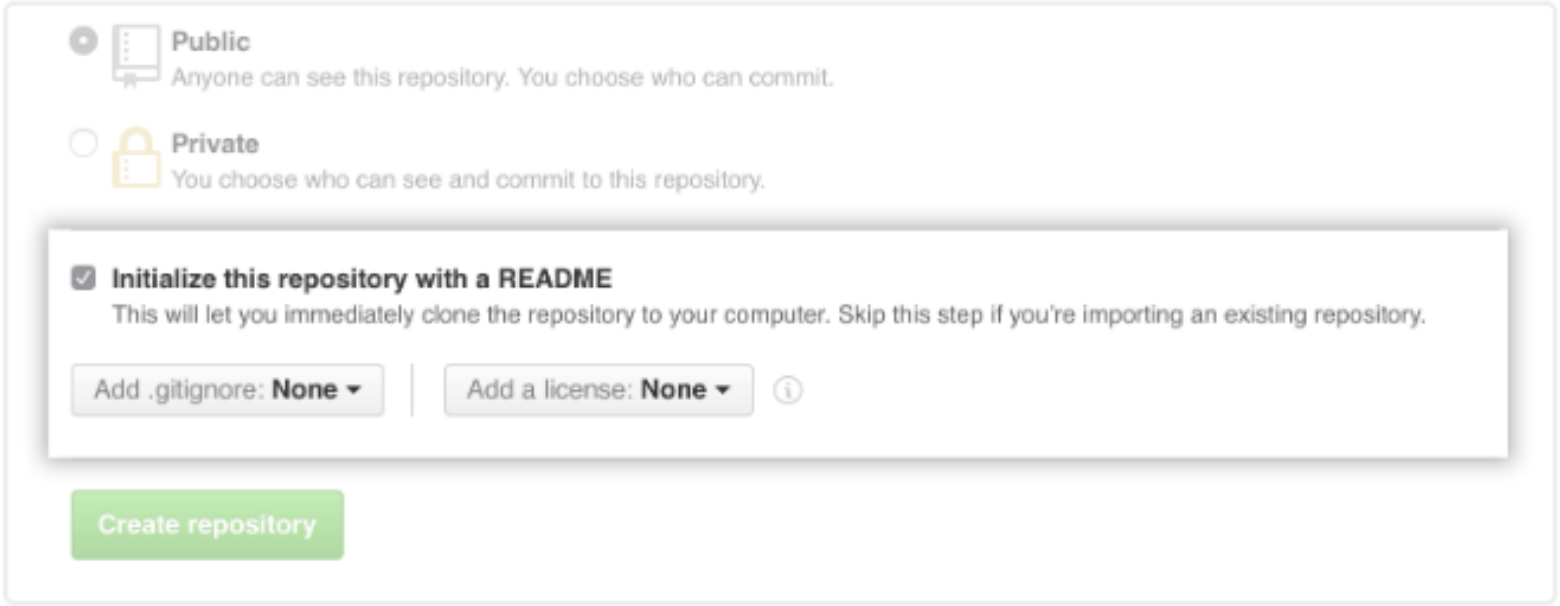
3 Optionally, add a description of your repository. For example, "My first repository on GitHub."



4 Choose to make the repository either public or private. Public repositories are visible to the public, while private repositories are only accessible to you, and people you share them with.



5 Select **Initialize this repository with a README.**



6 Click **Create repository.**

Congratulations! You've successfully created your first repository, and initialized it with a *README* file.

Setting repository visibility

About repository visibility

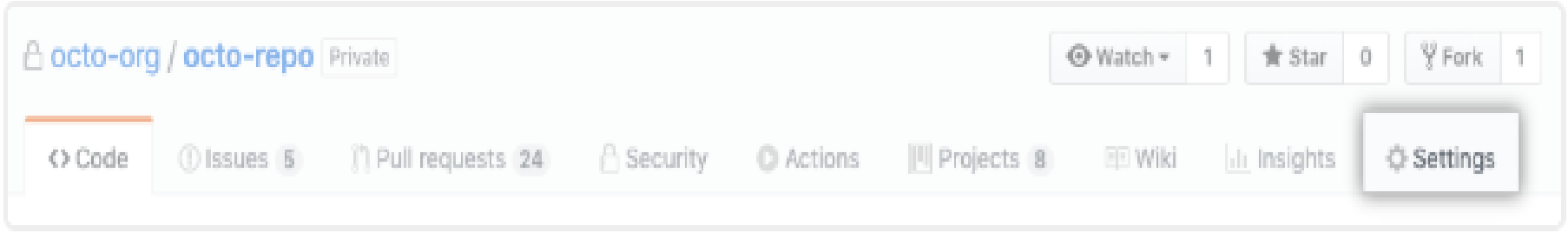
When creating a repository, you can choose to make it public, internal, or private. Public repositories are accessible to everyone using GitHub.com, while private repositories are accessible to you and the people you share them with.

Making a repository private

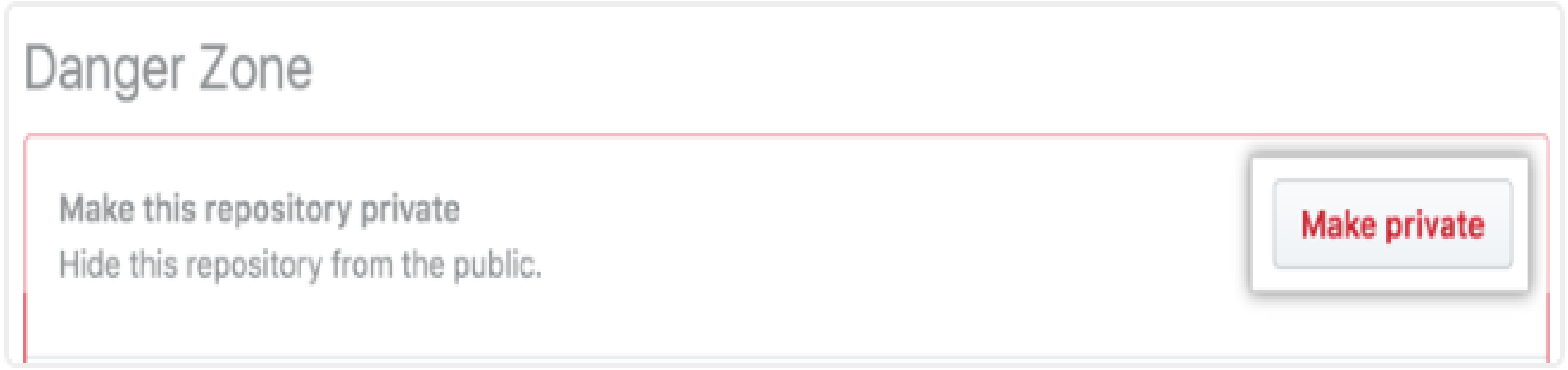
If you're using GitHub Free, private repositories owned by a personal account may have up to three collaborators. If you have added more than three other users as repository collaborators, you'll need to reduce the number of collaborators to three or fewer before making the repository private, or upgrade to GitHub Pro.

1 On GitHub, navigate to the main page of the repository.

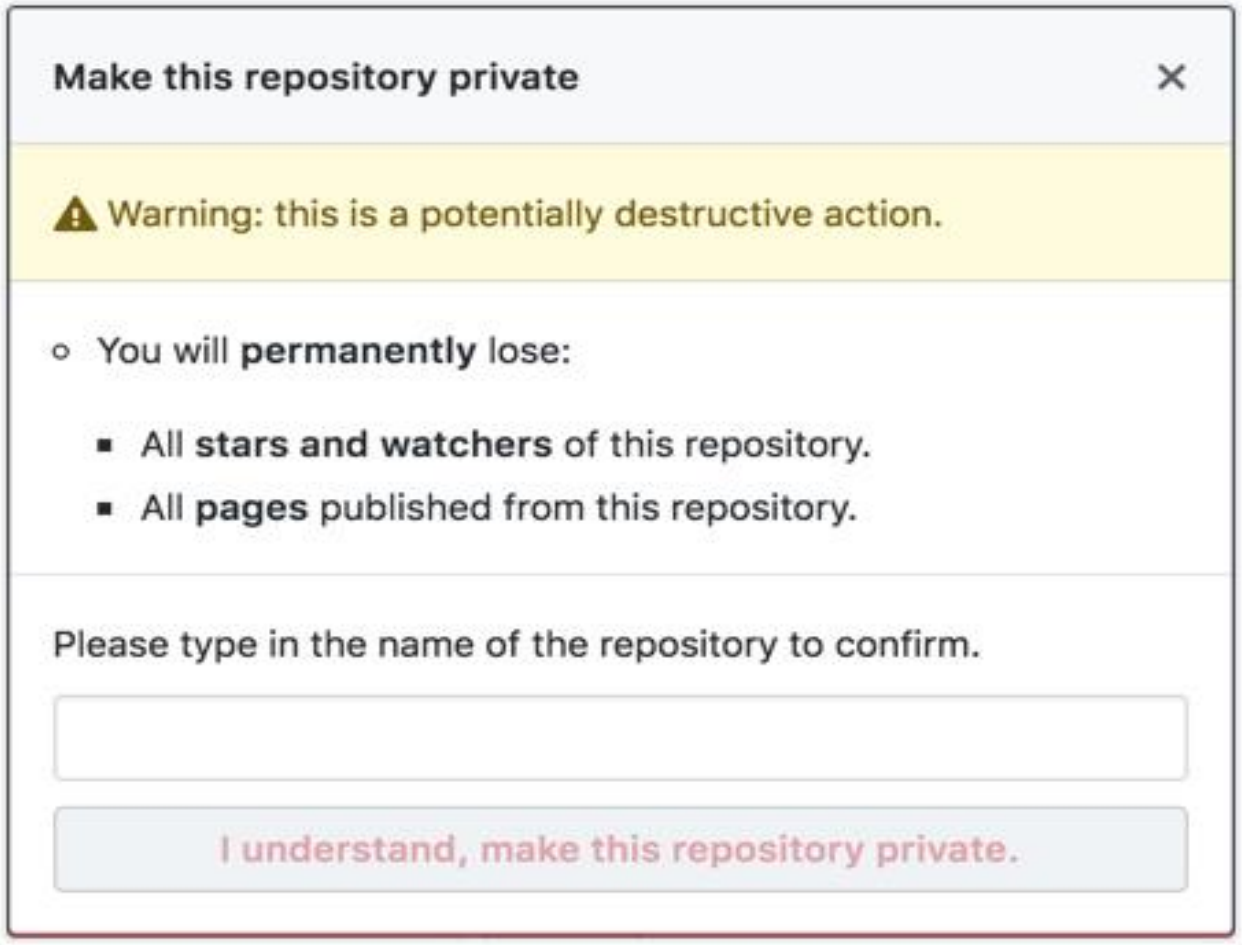
2 Under your repository name, click ⚙ **Settings**.



3 Under "Danger Zone", next to "Make this repository private", click **Make private.**



4 Read the warnings about making a repository private.

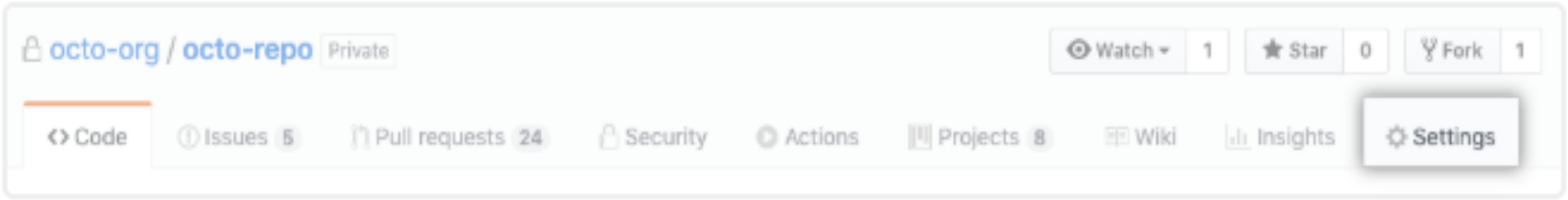


5 Type the name of the repository that you want to make private, for example `accountname/reponame`.

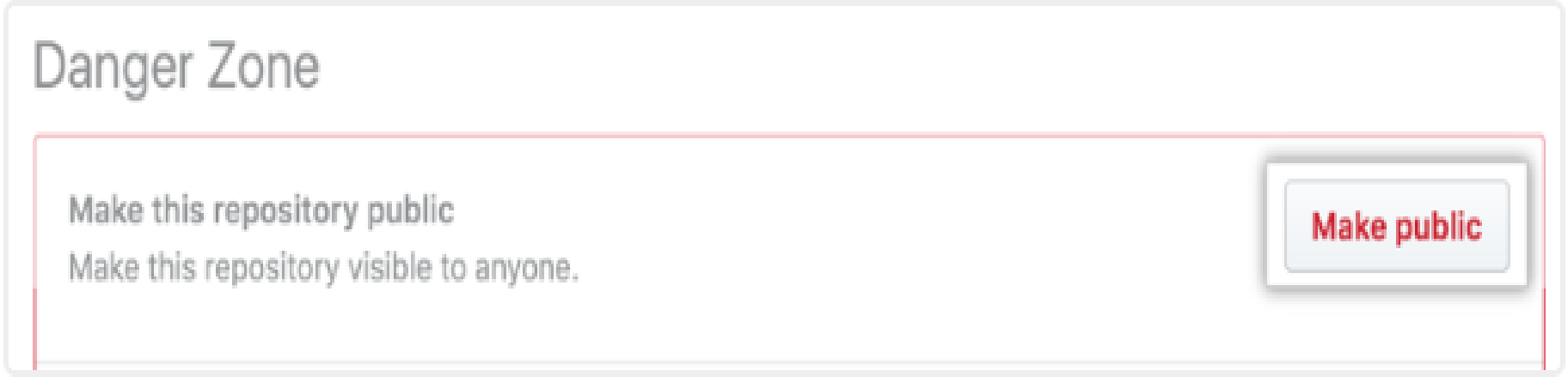
6 Click **I understand, make this repository private.**

Making a repository public

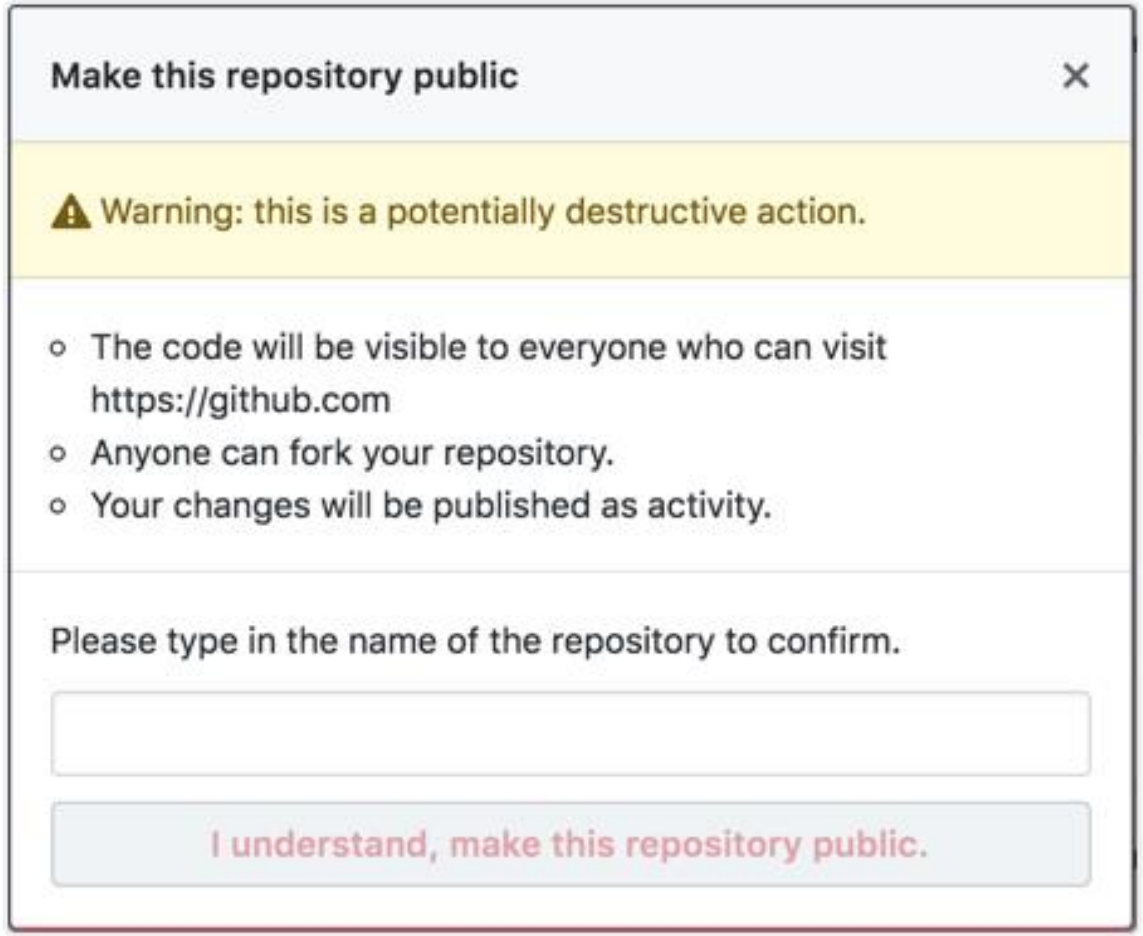
1. On GitHub, navigate to the main page of the repository.
2. Under your repository name, click ⚙ **Settings**.



3 Under "Danger Zone", next to "Make this repository public", click **Make public.**



4 Read the warnings about making a repository public.



5 Type the name of the repository that you want to make public, for example `accountname/reponame`.

6 Click I understand, make this repository public.

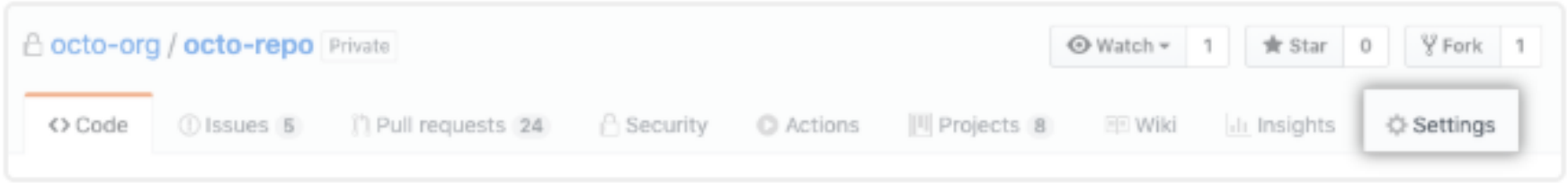
Inviting collaborators to a personal repository

You can invite users to become collaborators to your personal repository.

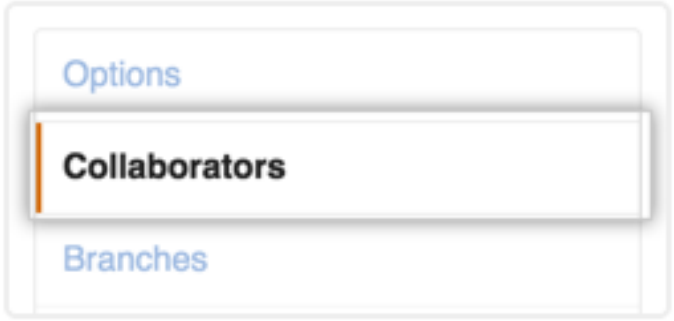
1. Ask for the username of the person you're inviting as a collaborator. If they don't have a username yet, they can sign up

2. On GitHub, navigate to the main page of the repository.

3 Under your repository name, click ⚙ **Settings**.

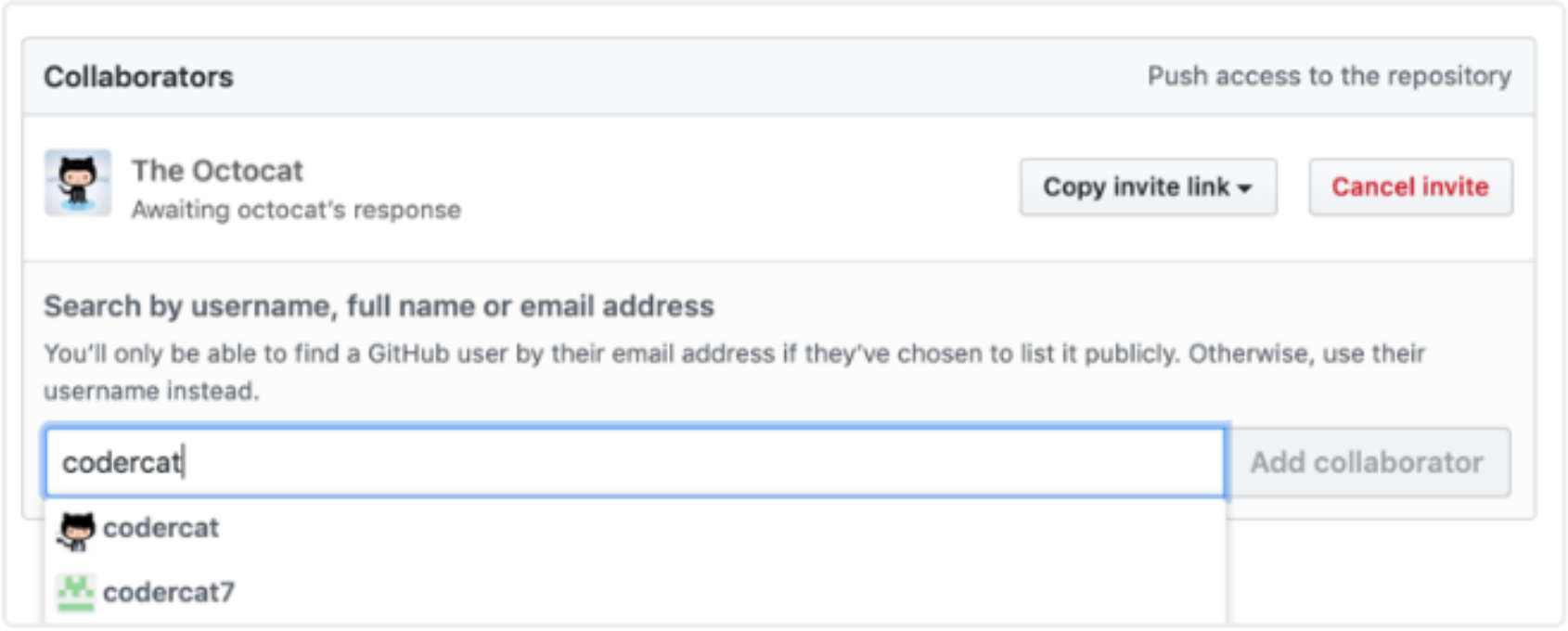


4 In the left sidebar, click **Collaborators**.

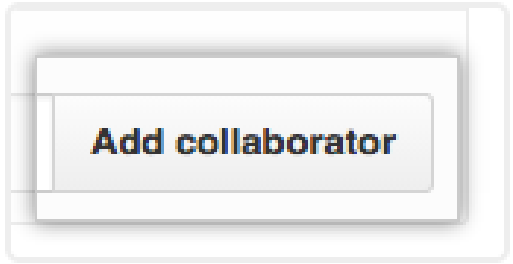


5 Under "Collaborators", start typing the collaborator's username.

6 Select the collaborator's username from the drop-down menu.



7 Click **Add collaborator**.



8 The user will receive an email inviting them to the repository. Once they accept your invitation, they will have collaborator access to your repository.

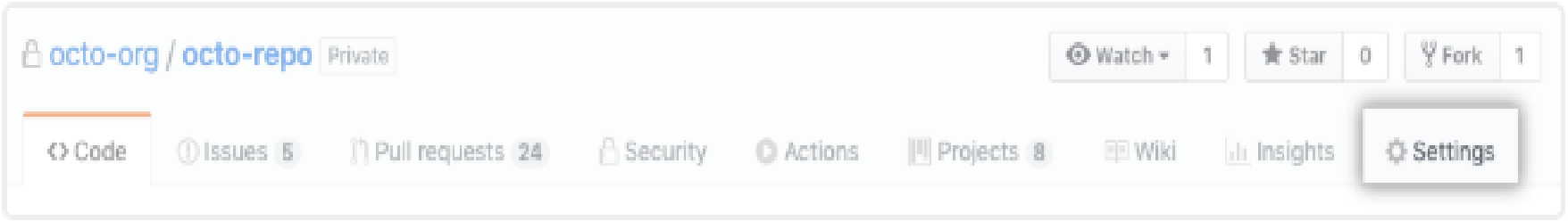
Removing a collaborator from a personal repository

If you're using GitHub Free, you can add unlimited collaborators on public repositories, and up to three collaborators on private repositories owned by your personal account. To add more than three other people as repository collaborators, upgrade to GitHub Pro.

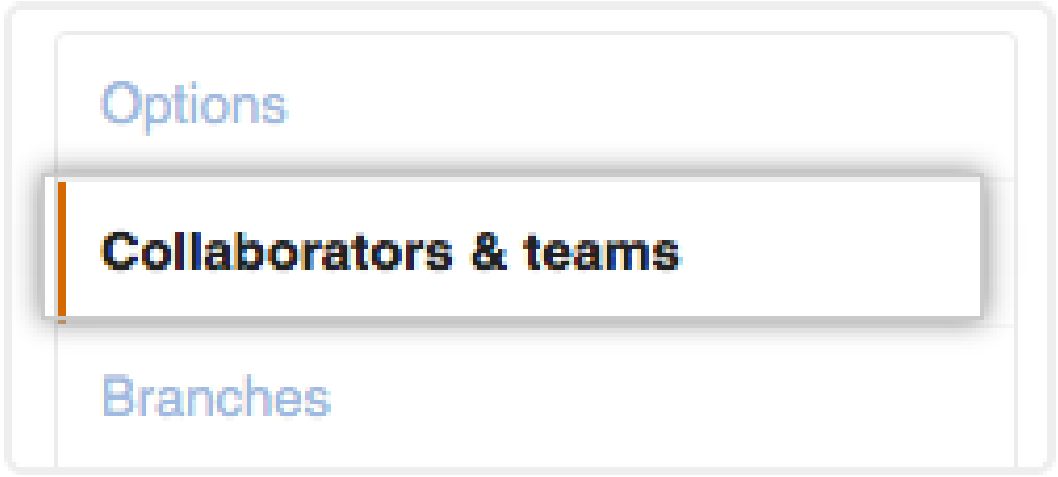
When you remove a collaborator from your project, they lose read/write access to your repository. If the repository is private and the person has created a fork, then that fork is also deleted.

Removing collaborator permissions from a person contributing to a repository

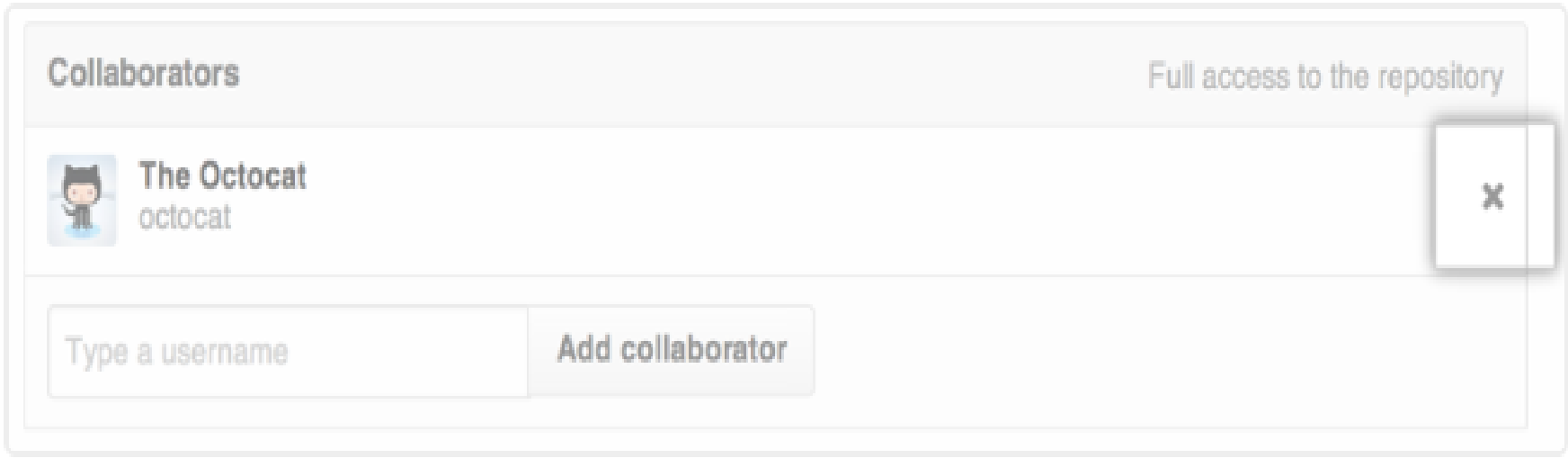
1. On GitHub, navigate to the main page of the repository.
2. Under your repository name, click ⚙ **Settings**.



3 In the left sidebar, click **Collaborators & teams**.



4 Next to the collaborator you want to remove, click the X icon.



Version control systems

- A version control system (often called a source code control system) does these things:
  - Keeps multiple (older and newer) versions of everything (not just source code)
  - Requests comments regarding every change
  - Allows “check in” and “check out” of files so you know which files someone else is working on
  - Displays differences between versions

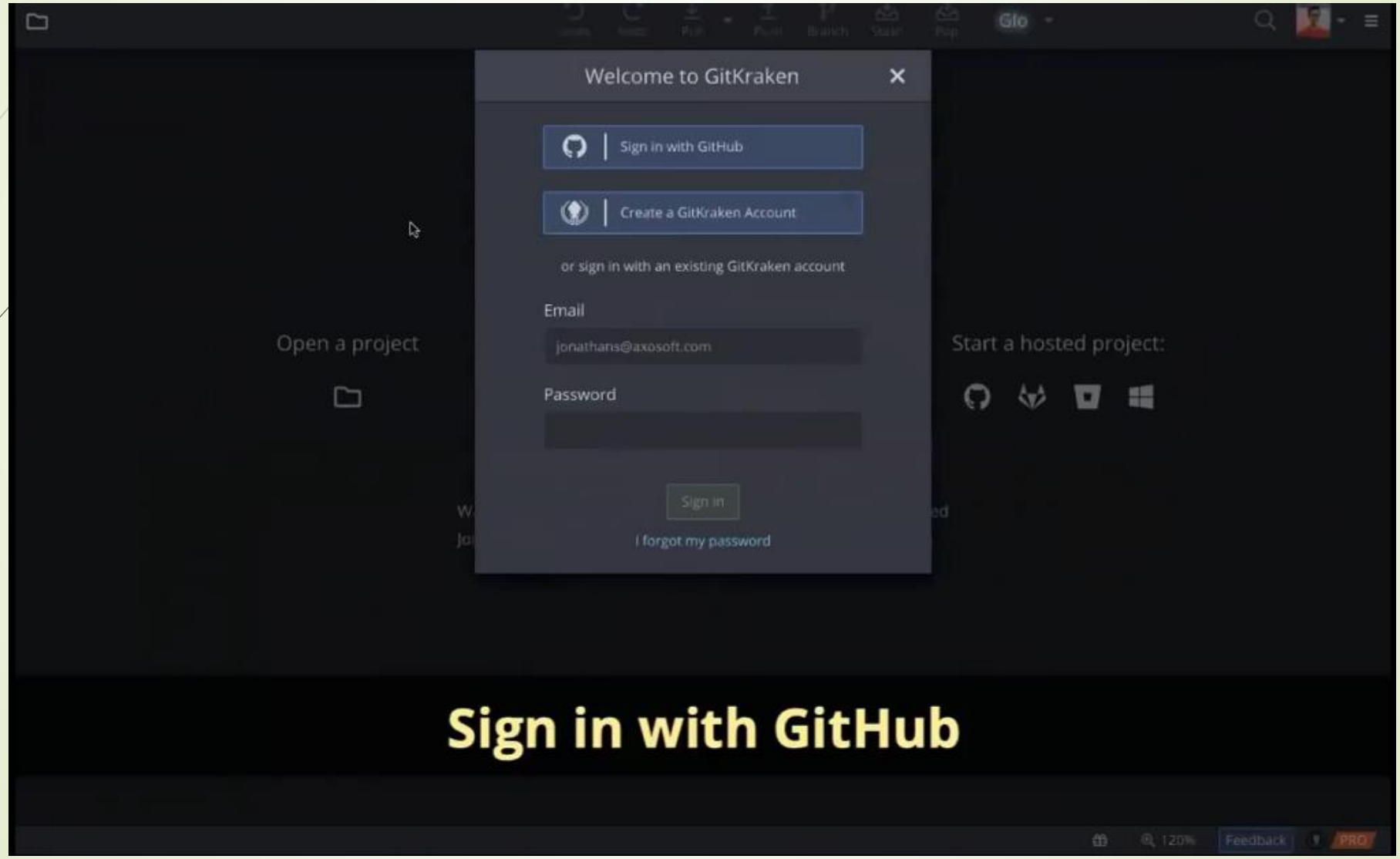
Benefits of version control

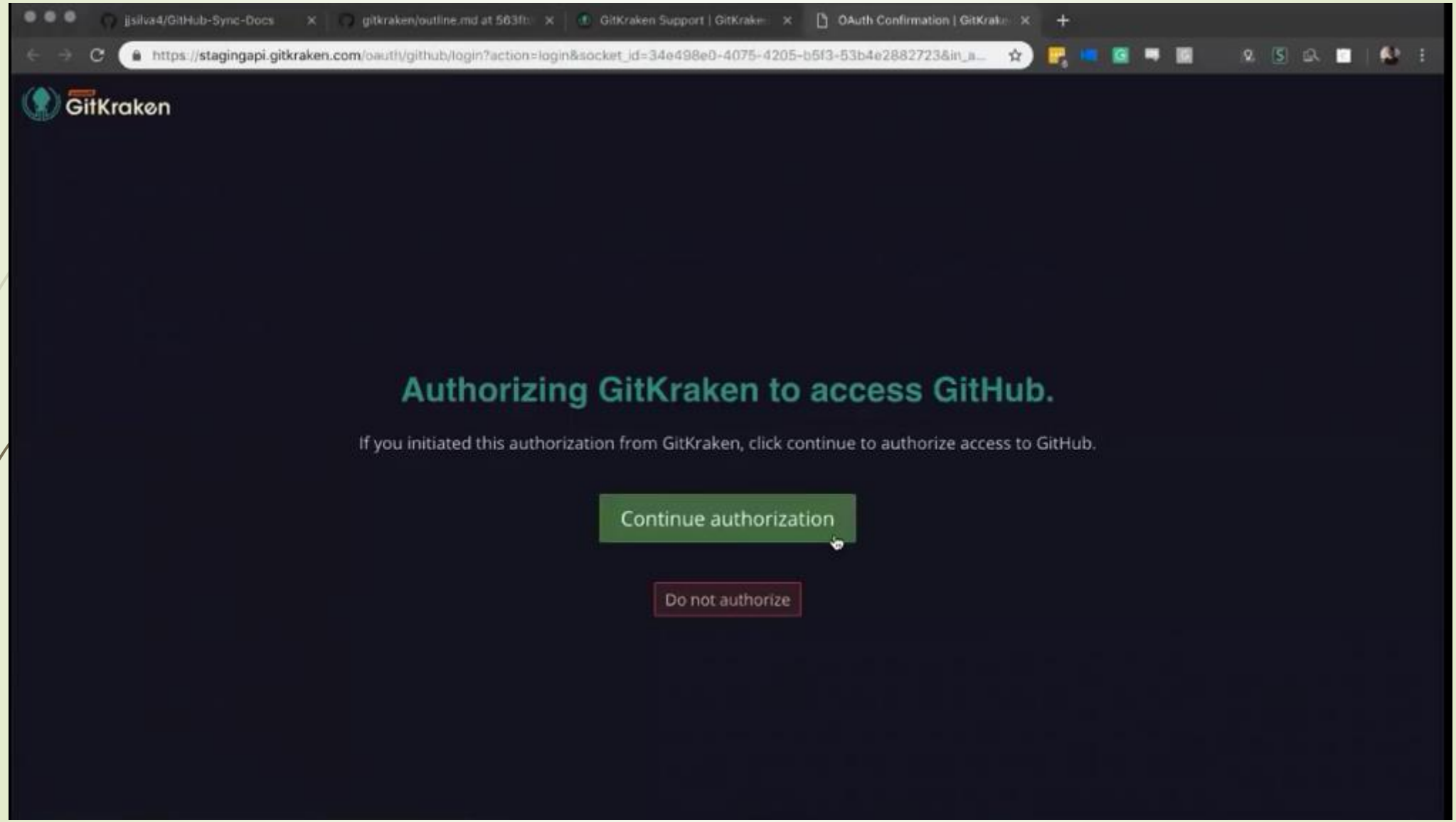
- For working by yourself:
  - Gives you a "time machine" for going back to earlier versions
  - Gives you great support for different versions (standalone, web app, etc.) of the same basic project
- For working with others:
  - Greatly simplifies concurrent work, merging changes

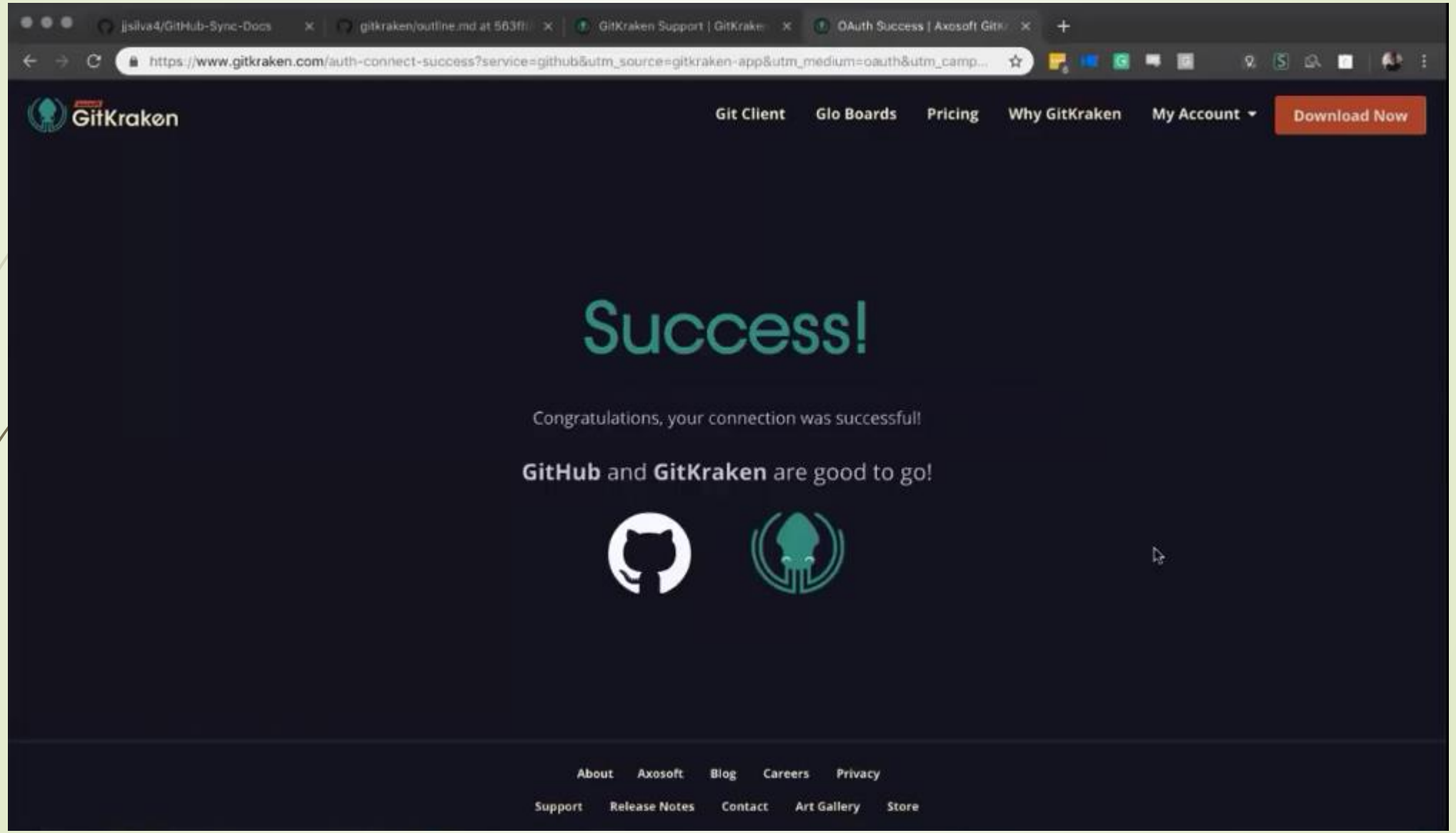
What are Git and GitHub

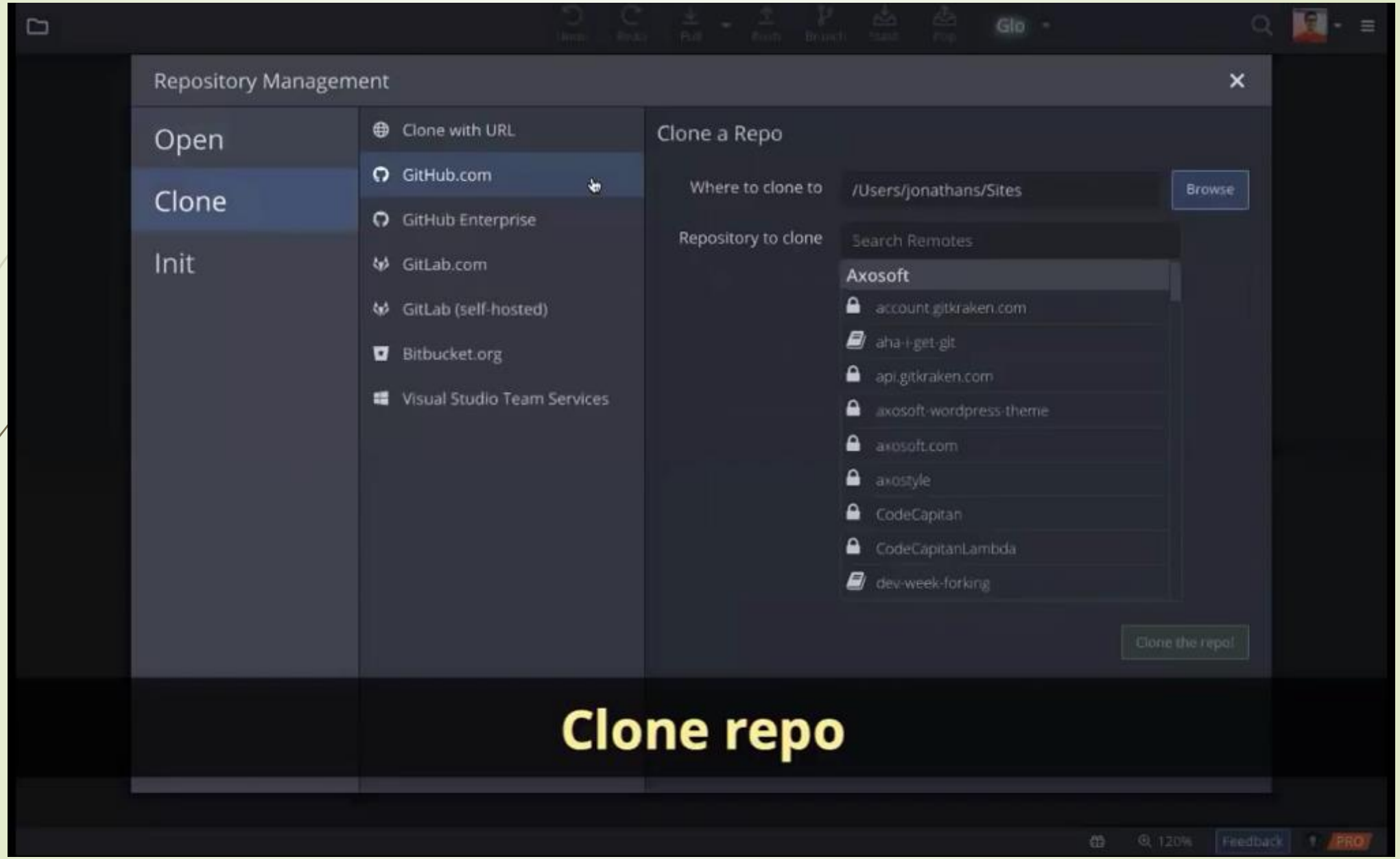
- Git is a free and open source distributed **version control system** designed to handle everything from small to very large projects with speed and efficiency
- GitHub is a **web-based** Git repository **hosting service**, which offers all of the distributed revision control and source code management (SCM) functionality of Git as well as adding its own features.

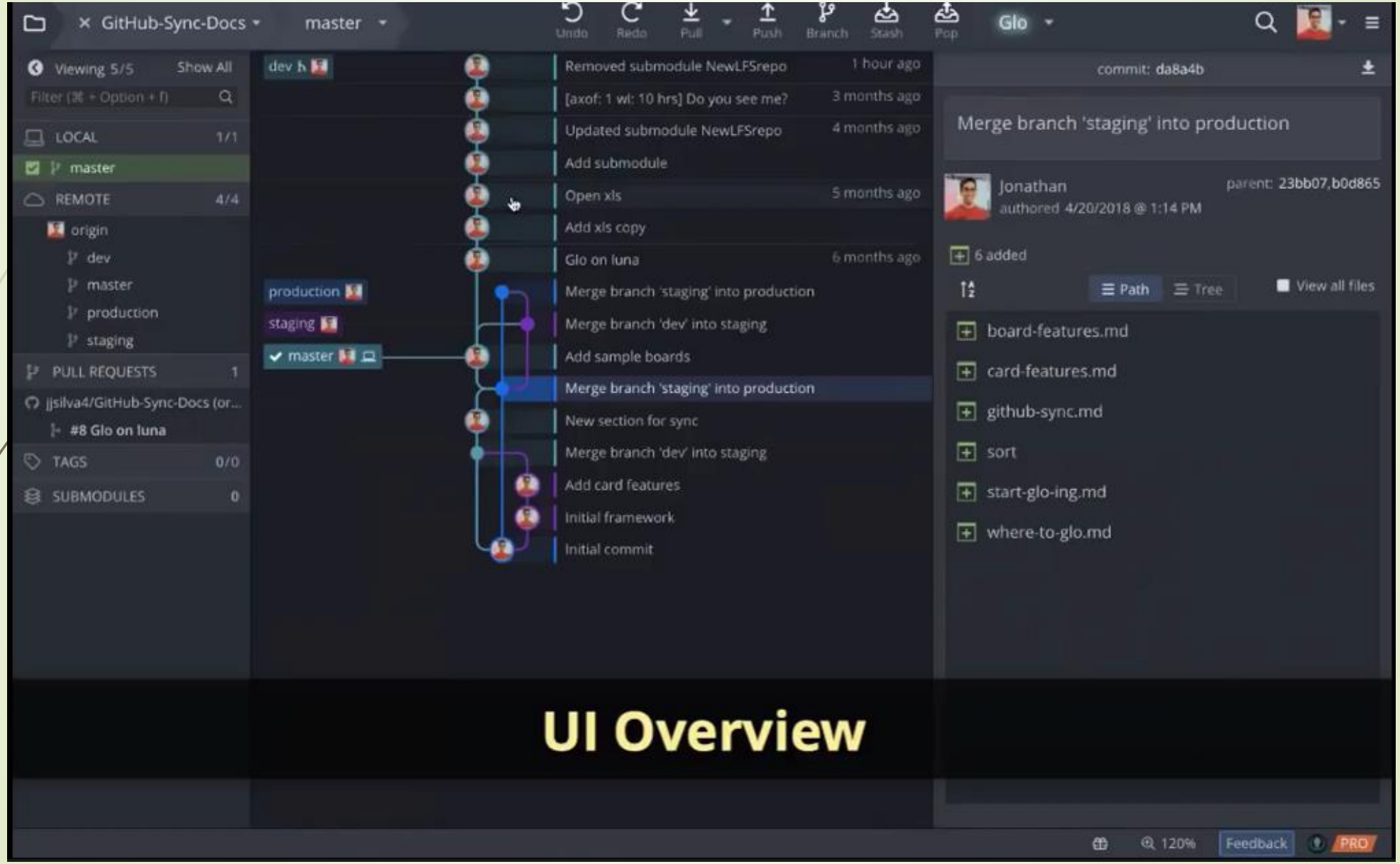
Gitkraken


Welcome to GitKraken
Sign in with GitHub
Create a GitKraken Account
or sign in with an existing GitKraken account
Email
jonathans@axosoft.com
Password
Sign in
I forgot my password
Open a project
Start a hosted project:
Sign in with GitHub
Feedback
PRO


jjsilva4/GitHub-Sync-Docs
gitkraken/outline.md at 563ft
GitKraken Support | GitKrake
OAuth Confirmation | GitKrak
https://stagingapi.gitkraken.com/oauth/github/login?action=login&socket_id=34e498e0-4075-4205-b5f3-53b4e2882723&in_a...
GitKraken
Authorizing GitKraken to access GitHub.
If you initiated this authorization from GitKraken, click continue to authorize access to GitHub.
Continue authorization
Do not authorize


jjsilva4/GitHub-Sync-Docs
gitkraken/outline.md at 563ff
GitKraken Support | GitKrake
OAuth Success | Axosoft Gitk
https://www.gitkraken.com/auth-connect-success?service=github&utm_source=gitkraken-app&utm_medium=oauth&utm_camp...
GitKraken
Git Client
Glo Boards
Pricing
Why GitKraken
My Account
Download Now
Success!
Congratulations, your connection was successful!
GitHub and GitKraken are good to go!
About
Axosoft
Blog
Careers
Privacy
Support
Release Notes
Contact
Art Gallery
Store


Repository Management
Open
Clone
Init
Clone with URL
GitHub.com
GitHub Enterprise
GitLab.com
GitLab (self-hosted)
Bitbucket.org
Visual Studio Team Services
Clone a Repo
Where to clone to
/Users/jonathans/Sites
Browse
Repository to clone
Search Remotes
Axosoft
account.gitkraken.com
aha-i-get-git
api.gitkraken.com
axosoft-wordpress-theme
axosoft.com
axostyle
CodeCapitan
CodeCapitanLambda
dev-week-forking
Clone the repo!
Clone repo


GitHub-Sync-Docs
master
Undo
Redo
Pull
Push
Branch
Stash
Pop
Glo
Viewing 5/5
Show All
Filter (⌘ + Option + f)
LOCAL 1/1
master
REMOTE 4/4
origin
dev
master
production
staging
PULL REQUESTS 1
jjsilva4/GitHub-Sync-Docs (or...
#8 Glo on luna
TAGS 0/0
SUBMODULES 0
dev
production
staging
master
Removed submodule NewLFSrepo 1 hour ago
[axof: 1 wl: 10 hrs] Do you see me? 3 months ago
Updated submodule NewLFSrepo 4 months ago
Add submodule
Open xls 5 months ago
Add xls copy
Glo on luna 6 months ago
Merge branch 'staging' into production
Merge branch 'dev' into staging
Add sample boards
Merge branch 'staging' into production
New section for sync
Merge branch 'dev' into staging
Add card features
Initial framework
Initial commit
commit: da8a4b
Merge branch 'staging' into production
Jonathan
authored 4/20/2018 @ 1:14 PM
parent: 23bb07,b0d865
6 added
Path
Tree
View all files
board-features.md
card-features.md
github-sync.md
sort
start-glo-ing.md
where-to-glo.md
120%
Feedback
PRO
UI Overview