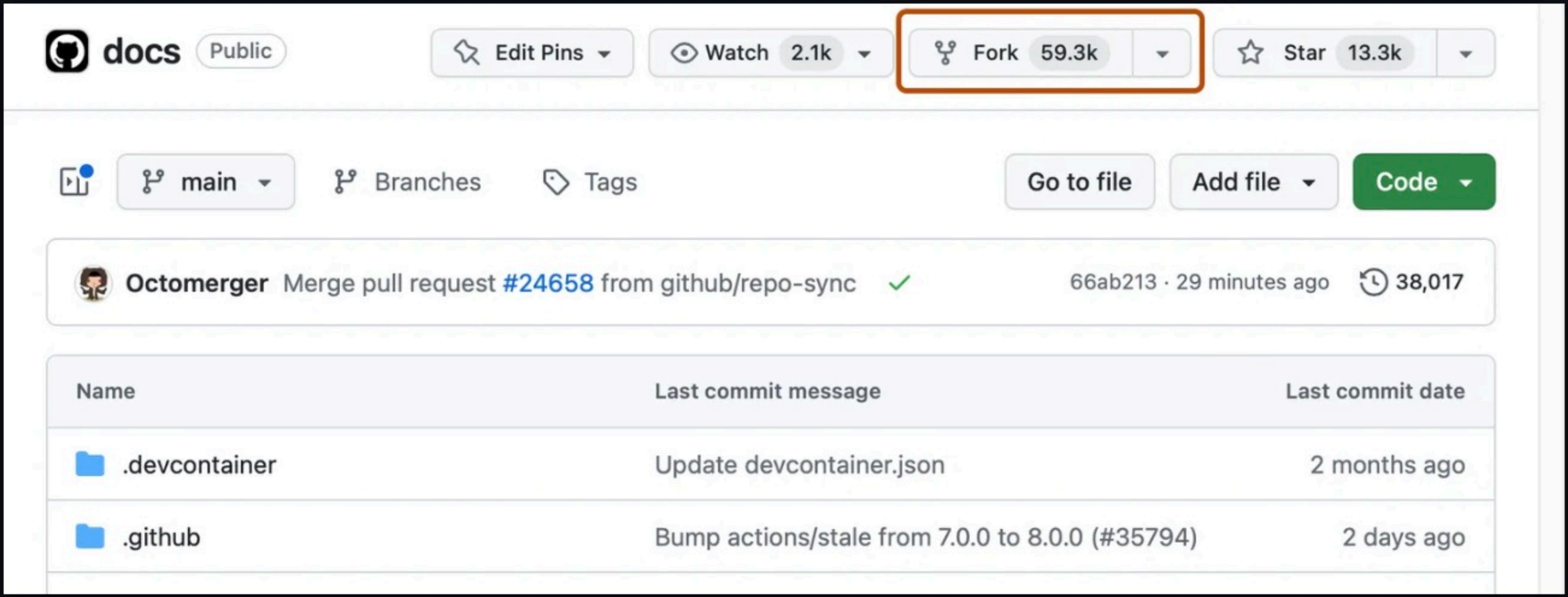
Forking a repository

This tutorial uses the Spoon-Knife project, a test repository that's hosted on GitHub.com that lets you test the fork and pull request workflow.

1. Navigate to the `Spoon-Knife` project at https://github.com/octocat/Spoon-Knife.
2. In the top-right corner of the page, click **Fork**.



Octomerger Merge pull request #24658 from github/repo-sync ✓ 66ab213 · 29 minutes ago 38,017

Name	Last commit message	Last commit date
.devcontainer	Update devcontainer.json	2 months ago
.github	Bump actions/stale from 7.0.0 to 8.0.0 (#35794)	2 days ago

3. Under "Owner," select the dropdown menu and click an owner for the forked repository.
4. By default, forks are named the same as their upstream repositories. Optionally, to further distinguish your fork, in the "Repository name" field, type a name.
5. Optionally, in the "Description" field, type a description of your fork.

5. Optionally, in the "Description" field, type a description of your fork.

6. Optionally, select **Copy the DEFAULT branch only**.

   For many forking scenarios, such as contributing to open-source projects, you only need to copy the default branch. If you do not select this option, all branches will be copied into the new fork.

7. Click **Create fork**.

> **Note:** If you want to copy additional branches from the upstream repository, you can do so from the **Branches** page. For more information, see "Creating and deleting branches within your repository."

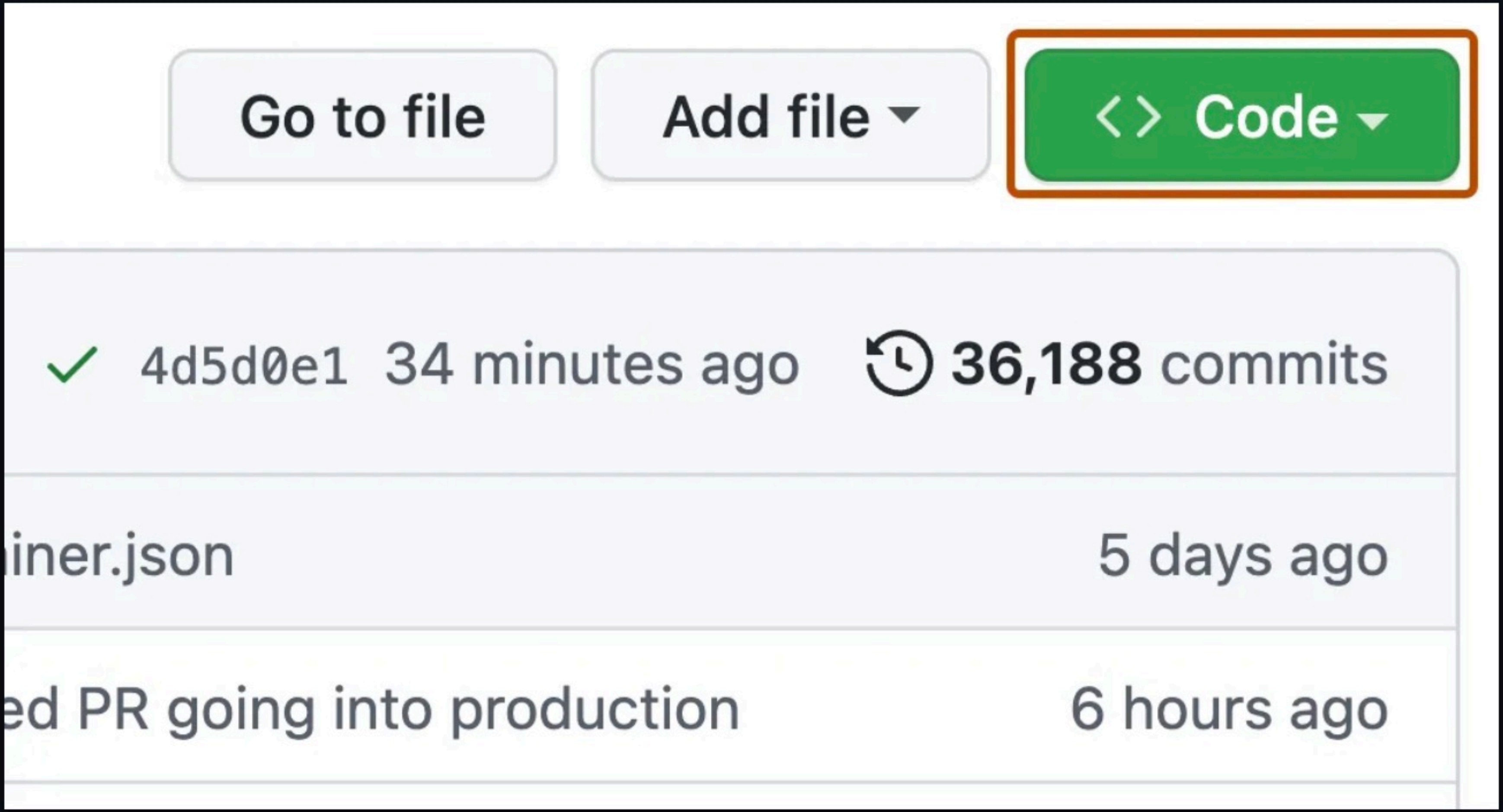
Cloning a fork

You've successfully forked the Spoon-Knife repository, but so far, it only exists on GitHub. To be able to work on the project, you will need to clone it to your computer.

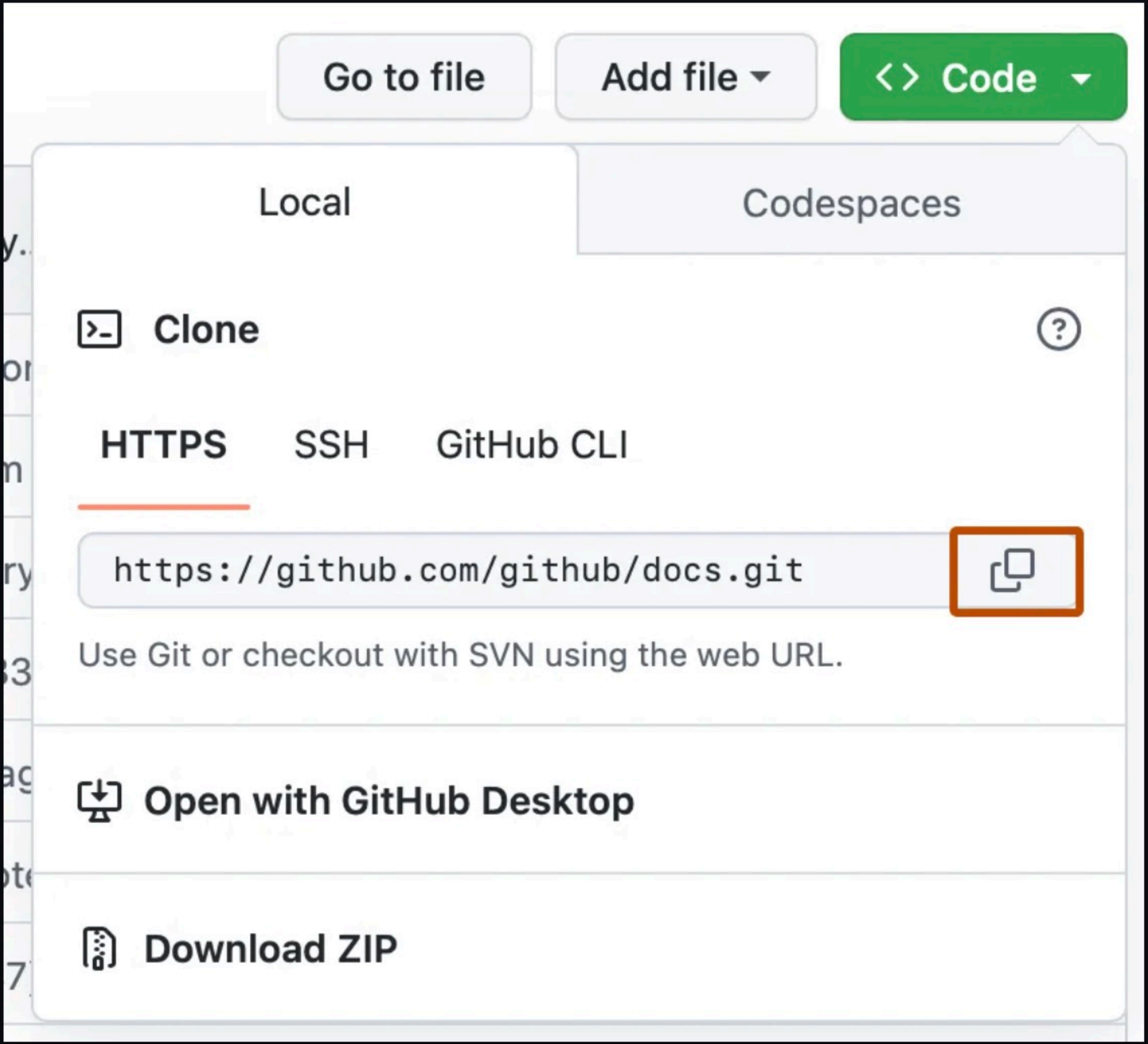
You can clone your fork with the command line, GitHub CLI, or GitHub Desktop.

1. On GitHub, navigate to **your fork** of the Spoon-Knife repository.

2. Above the list of files, click <> **Code**.



3. Copy the URL for the repository.
   - To clone the repository using HTTPS, under "HTTPS", click 📋.
   - To clone the repository using an SSH key, including a certificate issued by your organization's SSH certificate authority, click **SSH**, then click 📋.
   - To clone a repository using GitHub CLI, click **GitHub CLI**, then click ⧉.



https://github.com/github/docs.git

Use Git or checkout with SVN using the web URL.

4. Open Terminal.

5. Change the current working directory to the location where you want the cloned directory.

6. Type `git clone`, and then paste the URL you copied earlier. It will look like this, with your GitHub username instead of `YOUR-USERNAME`:

```
git clone https://github.com/YOUR-USERNAME/Spoon-Knife
```

7. Press **Enter**. Your local clone will be created.

```
$ git clone https://github.com/YOUR-USERNAME/Spoon-Knife
> Cloning into `Spoon-Knife`...
> remote: Counting objects: 10, done.
> remote: Compressing objects: 100% (8/8), done.
> remove: Total 10 (delta 1), reused 10 (delta 1)
> Unpacking objects: 100% (10/10), done.
```

Creating a branch to work on

Before making changes to the project, you should create a new branch and check it out. By keeping changes in their own branch, you follow GitHub Flow and ensure that it will be easier to contribute to the same project again in the future. For more information, see "GitHub flow."

```
git branch BRANCH-NAME
git checkout BRANCH-NAME
```

Making and pushing changes

Go ahead and make a few changes to the project using your favorite text editor, like Visual Studio Code. You could, for example, change the text in `index.html` to add your GitHub username.

When you're ready to submit your changes, stage and commit your changes. `git add .` tells Git that you want to include all of your changes in the next commit. `git commit` takes a snapshot of those changes.

```
git add .
git commit -m "a short description of the change"
```

When you stage and commit files, you essentially tell Git, "Okay, take a snapshot of my changes!" You can continue to make more changes, and take more commit snapshots.

Right now, your changes only exist locally. When you're ready to push your changes up to GitHub, push your changes to the remote.

```
git push
```