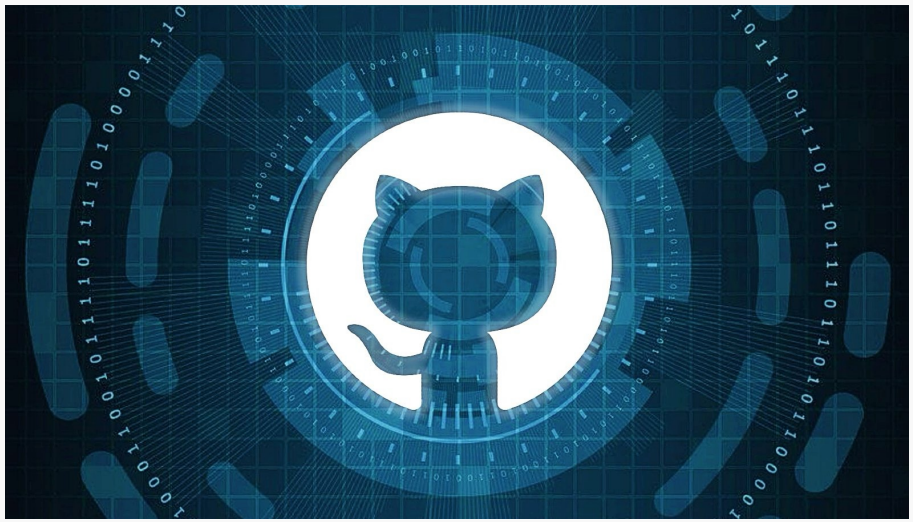
Presentation by Daniel Perez
Graduate Fellow
Murty Sunak Quantitative Computing Lab

GitHub Workflow Basics

Requirements

- GitHub desktop installed
- Account and Logged into GitHub
- Create a working directory in your computer



- Learn basic concepts of version control systems
- Learn how to setup a git repository and organization
- Learn how to setup a project and prepare to contribute to it by cloning and branching
- Learn how to create and assign issues
- Learn about pull requests and code review

101

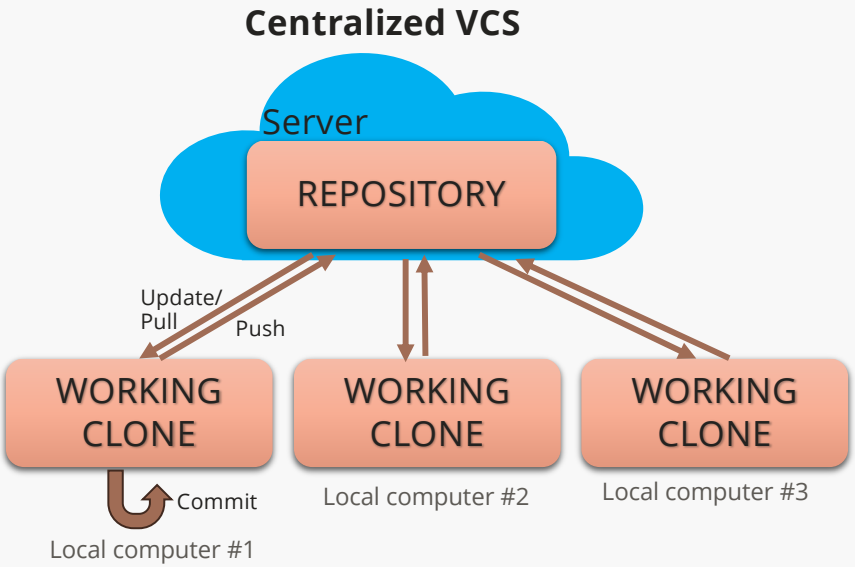
Version Control System:
Software tool that helps in recording changes made to files by keeping a track of modifications done in the code.

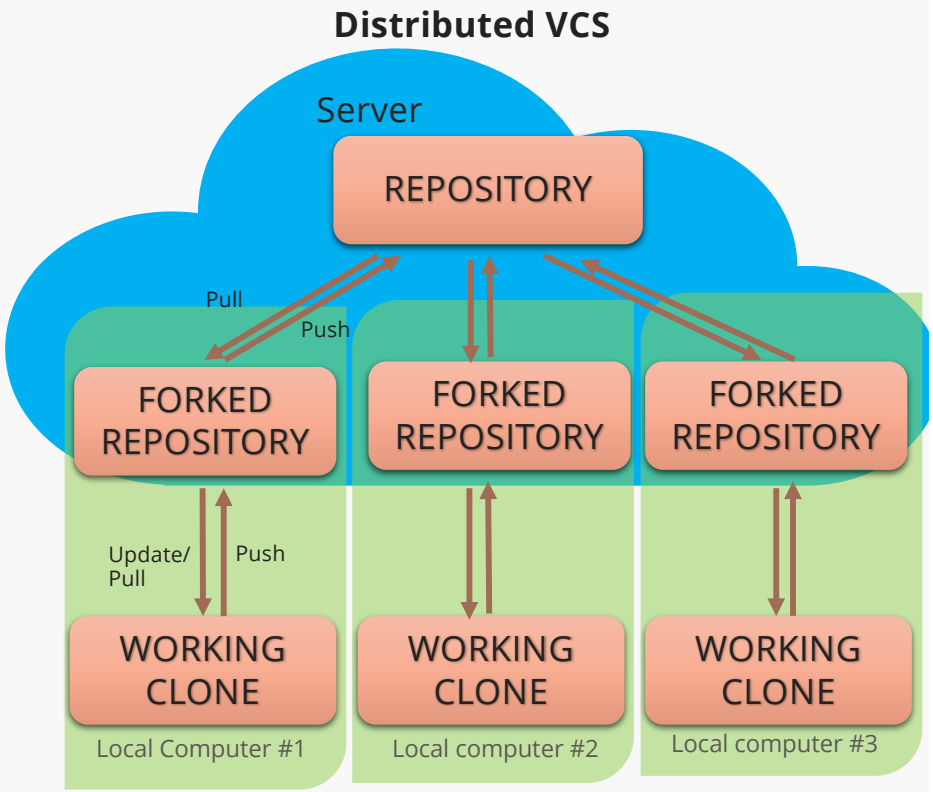
Why do we use a Version Control System (VCS)?

- Keep a complete change history of every file; the change history contains essential information such as which change was made, who made the change, when the change happened, and why the change was made.
- Let people work together independently at the same time
- Make project management easier (traceability, reversibility, backup)

Types of VCS

- Local Version Control Systems
- Centralized Version Control Systems
- Distributed Version Control Systems





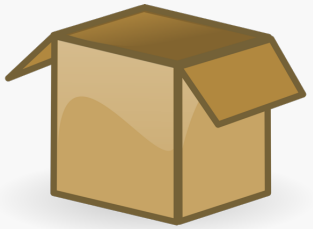
What is GitHub?

GitHub is a web-based platform used for version control, collaboration, and code repository management. It allows developers to work together on projects, track changes, merge code, and manage issues.

Key Components

Repositories

Repositories in GitHub are where your project's files and revision history are stored. They can be public or private, and each repository contains documentation, code, and related resources.



Issues

Issues in GitHub are used to track tasks, enhancements, and bugs for your projects. They facilitate communication among team members, assign tasks, and prioritize work.

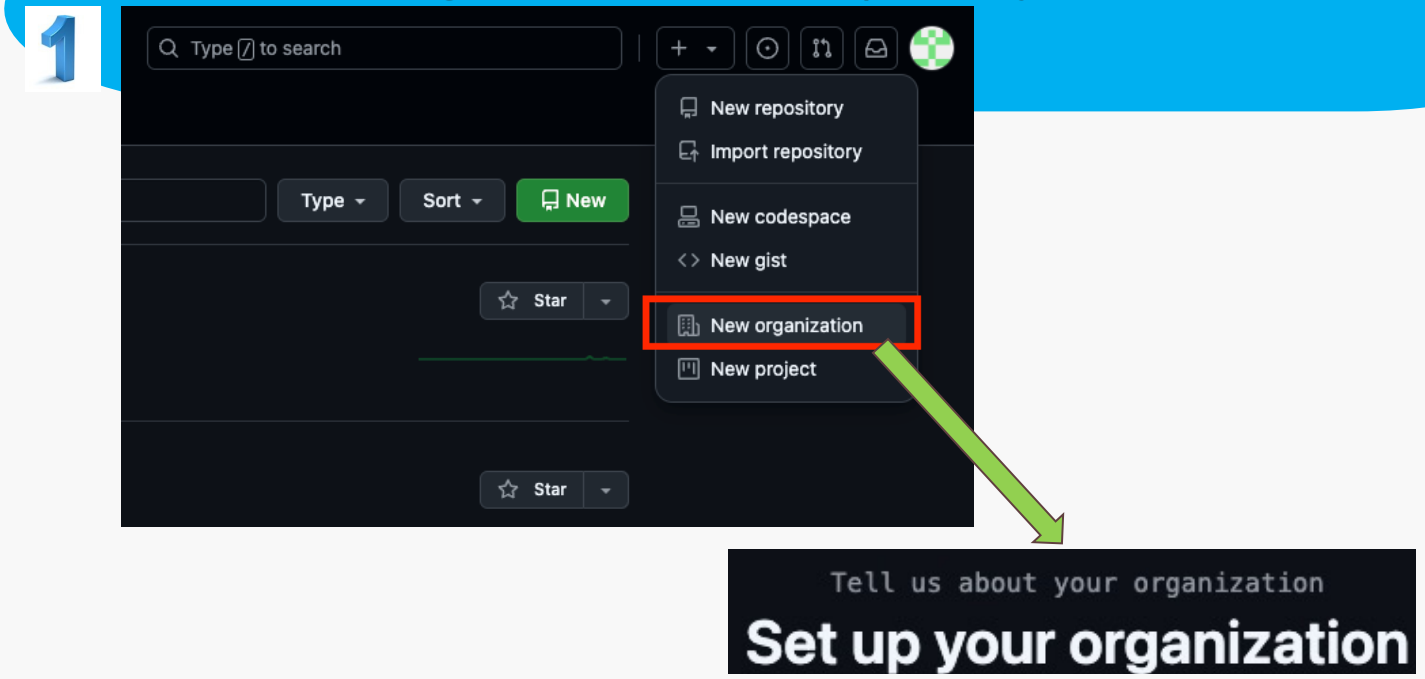
Roles: Maintainer/Contributor

The Maintainer as the person who is owner and in charge of the main repository, manage a project's direction and improvement

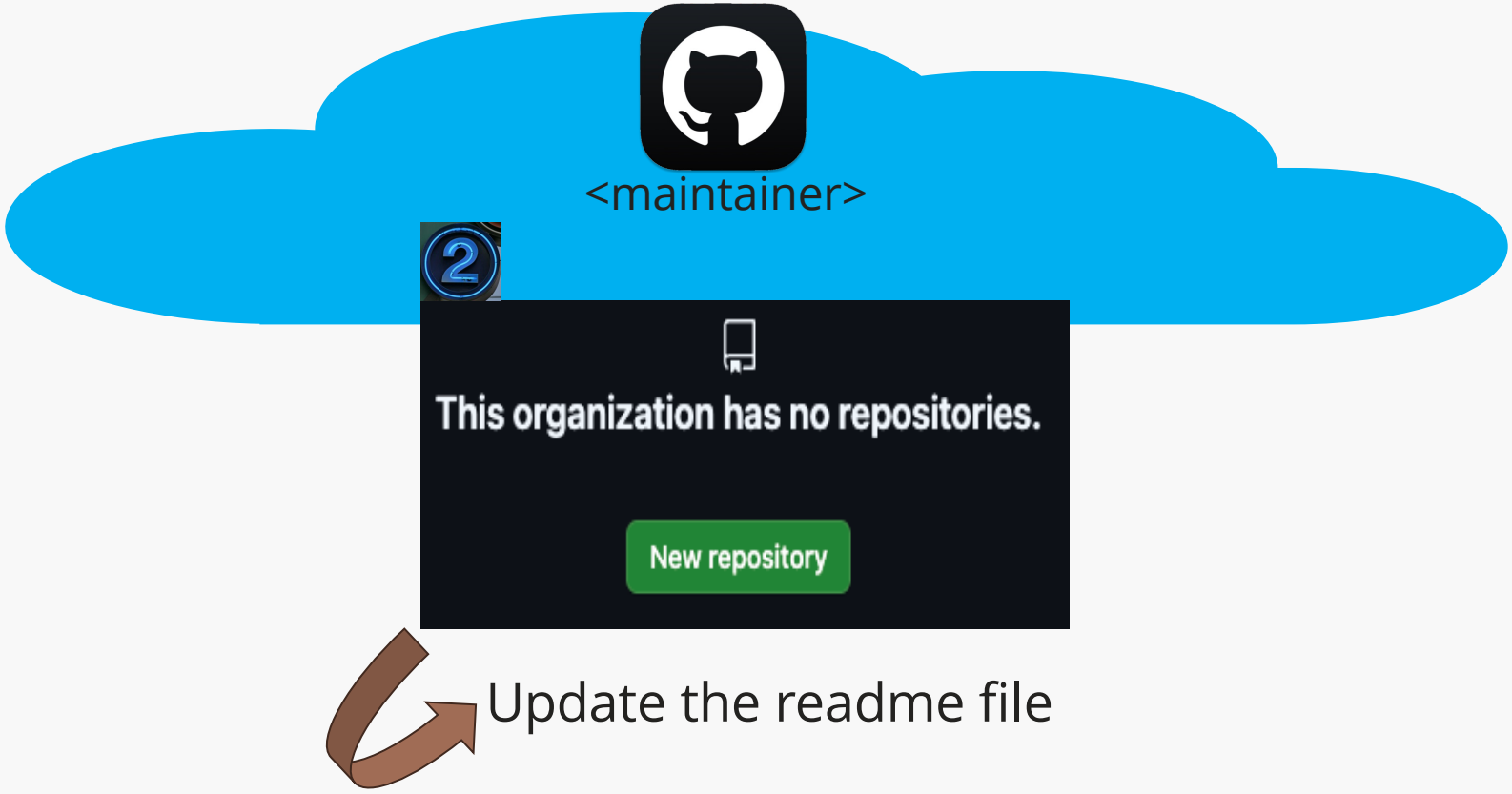
The Contributor as the person who is accessing or helping modify the repository
Roles: Read, Triage, Write, Admin

Prepare the project <Maintainer>

Creates an organization and a repository



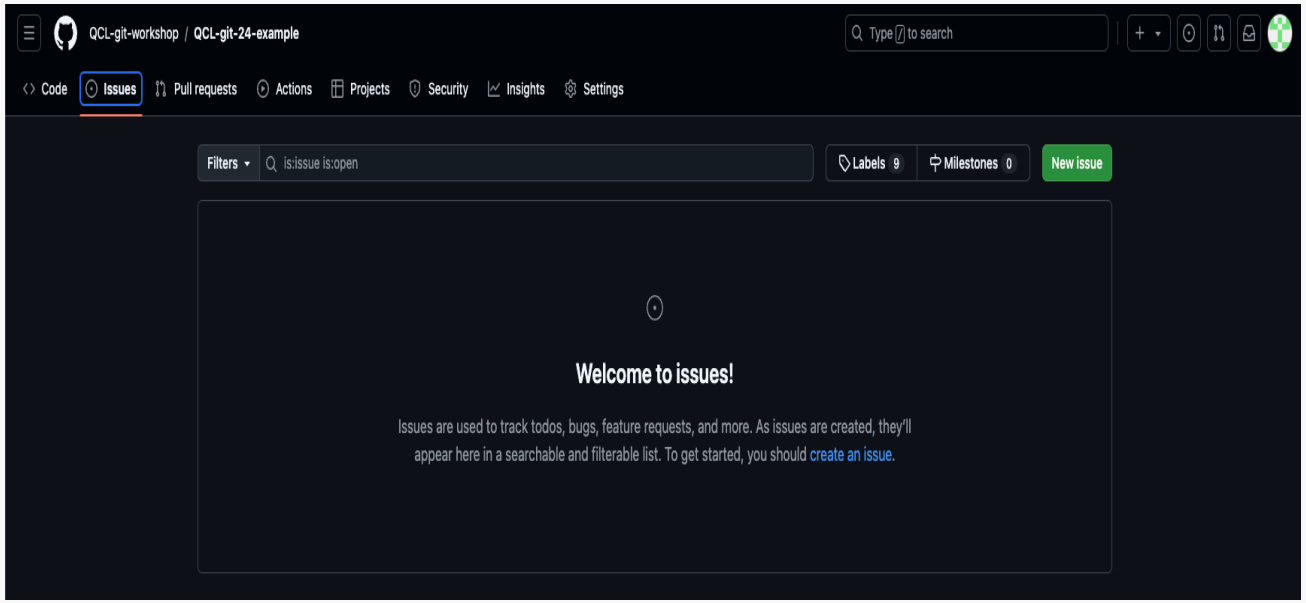
Prepare the project



Issues and collaborations <Maintainer>

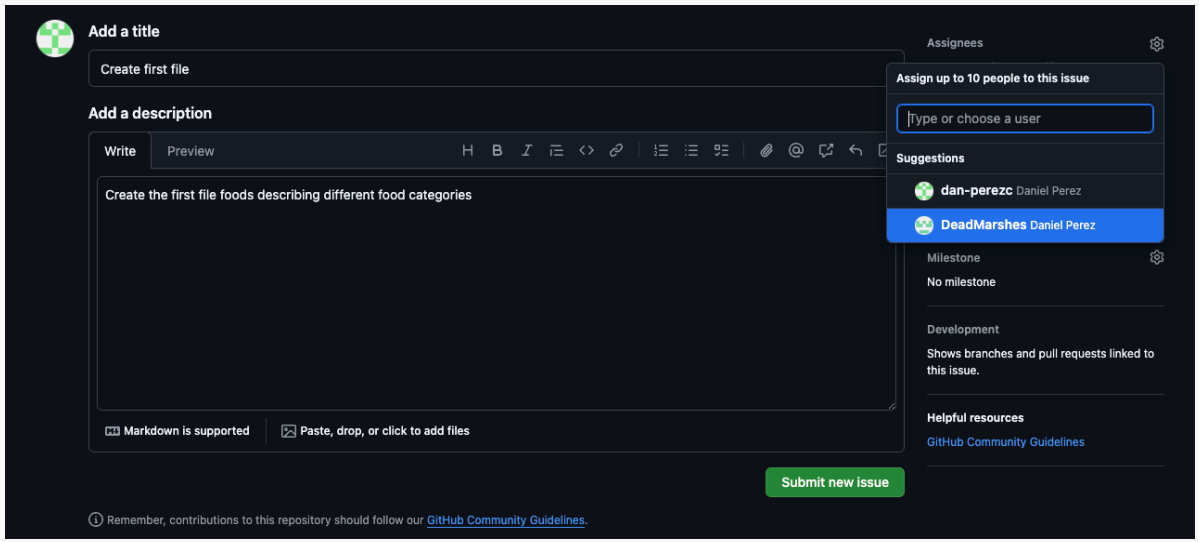
Creating and issue and then claiming it

Helps prevent developers from working on the same thing at the same time. Also, issue provides a place for the community to propose and ideas, prioritize issues, size issues, clarify requirements, and verify bugs



Issues and collaborations

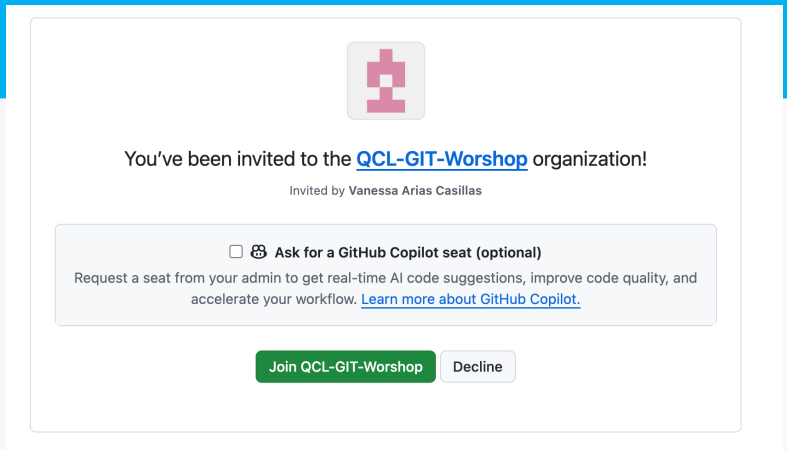
Create and assign an
issue to your Contributor

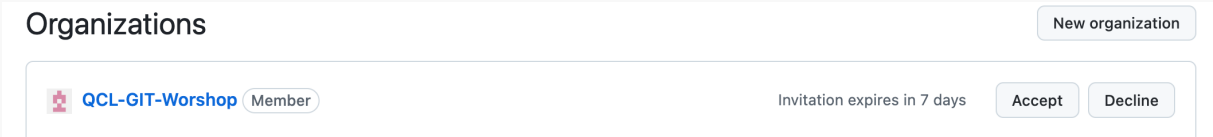


Join to the project < Contributor >

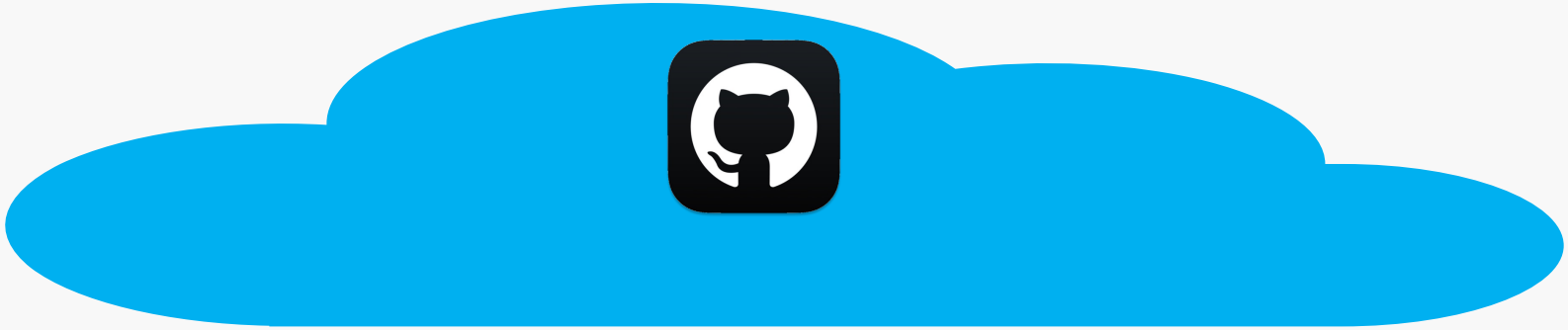
Receives invitation to the company or project

Either
Find the invitation in your email, view the invitation, and accept the invitation.
Or
Navigate to the team's organization, view the invitation, and accept the invitation.



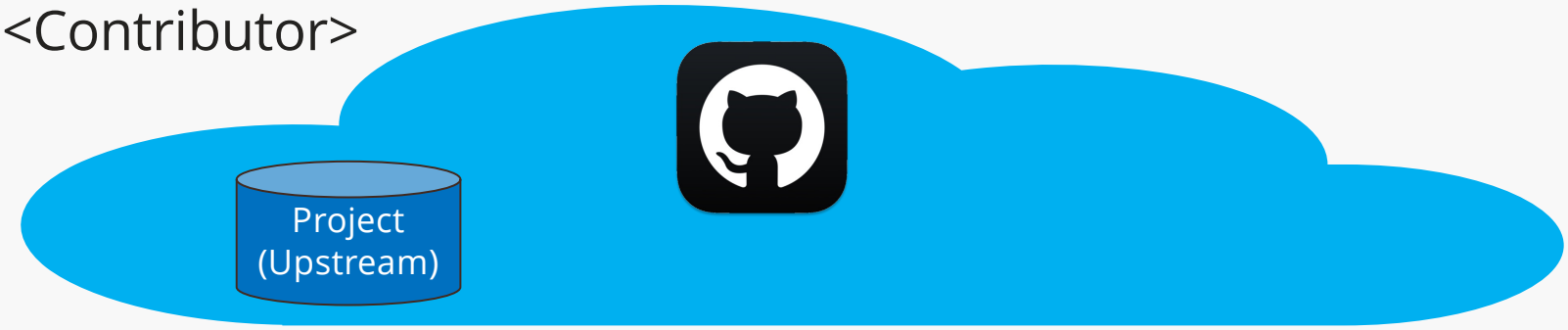


Project tracker



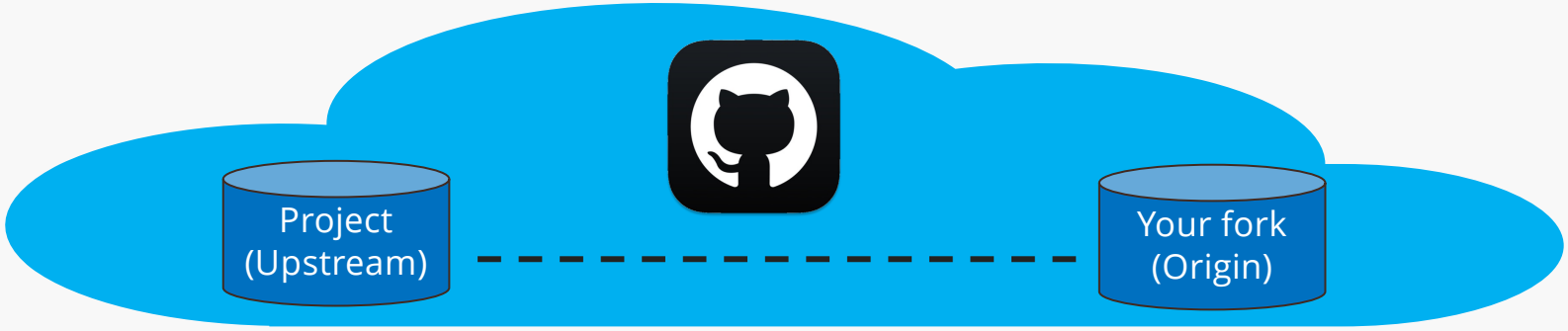
Branch/TAG	GRAPH	COMMIT	USER
Main/master		Update on file Readme	Maintainer

Prepare to work on a project

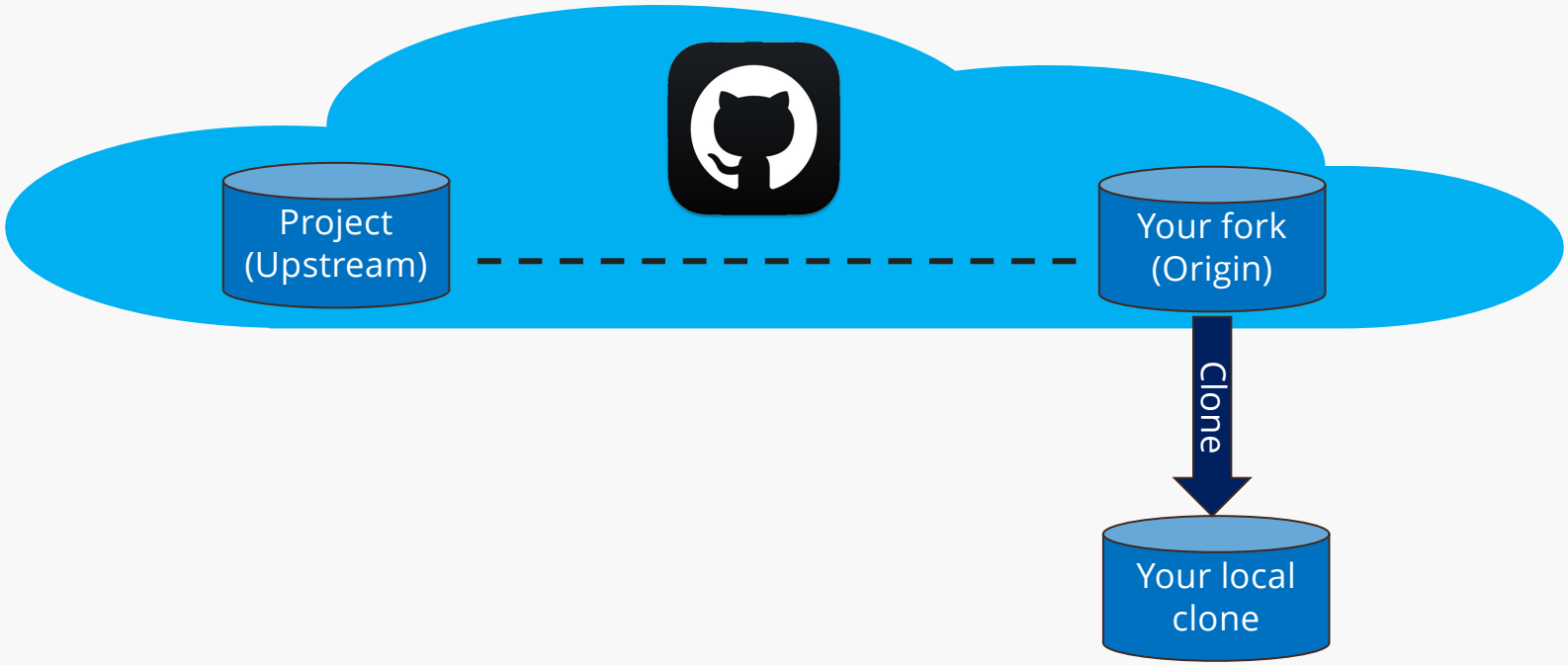


Prepare to work on a project

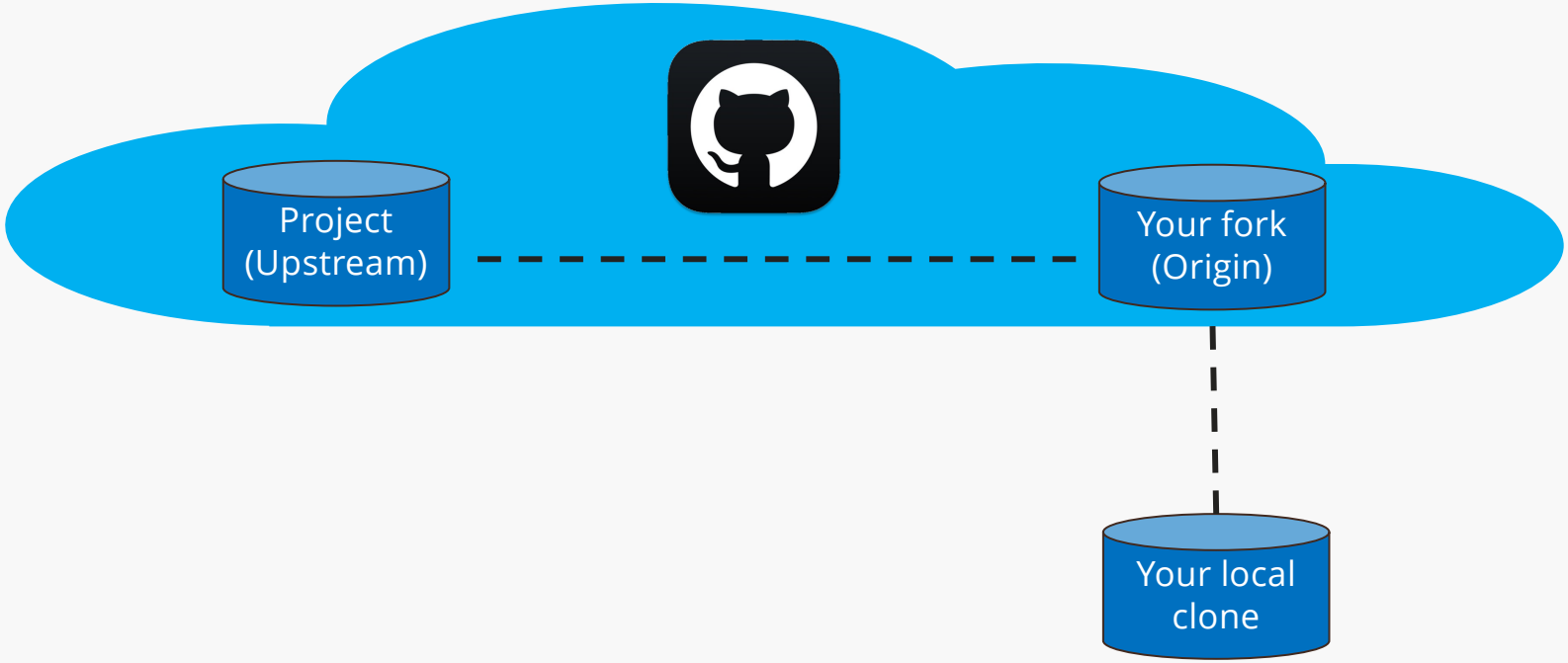
Prepare to work on a project



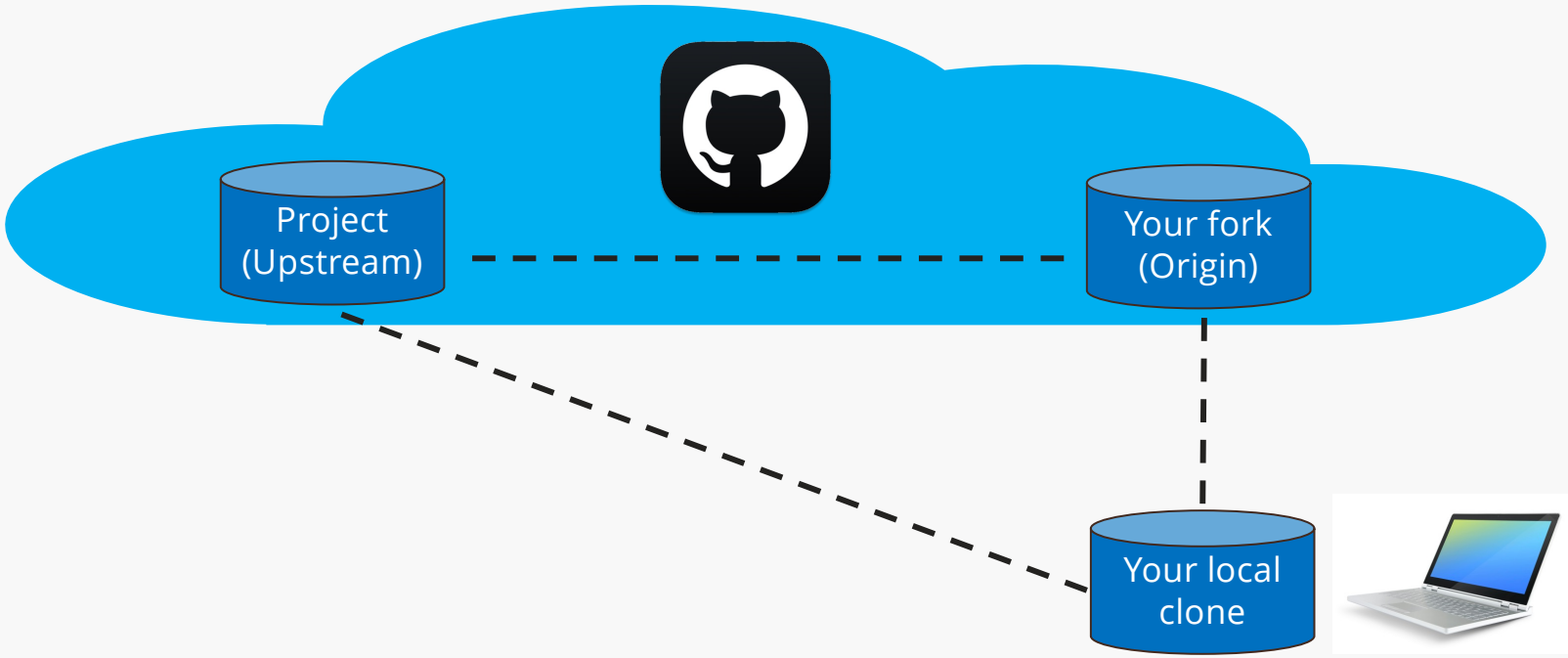
Prepare to work on a project



Prepare to work on a project

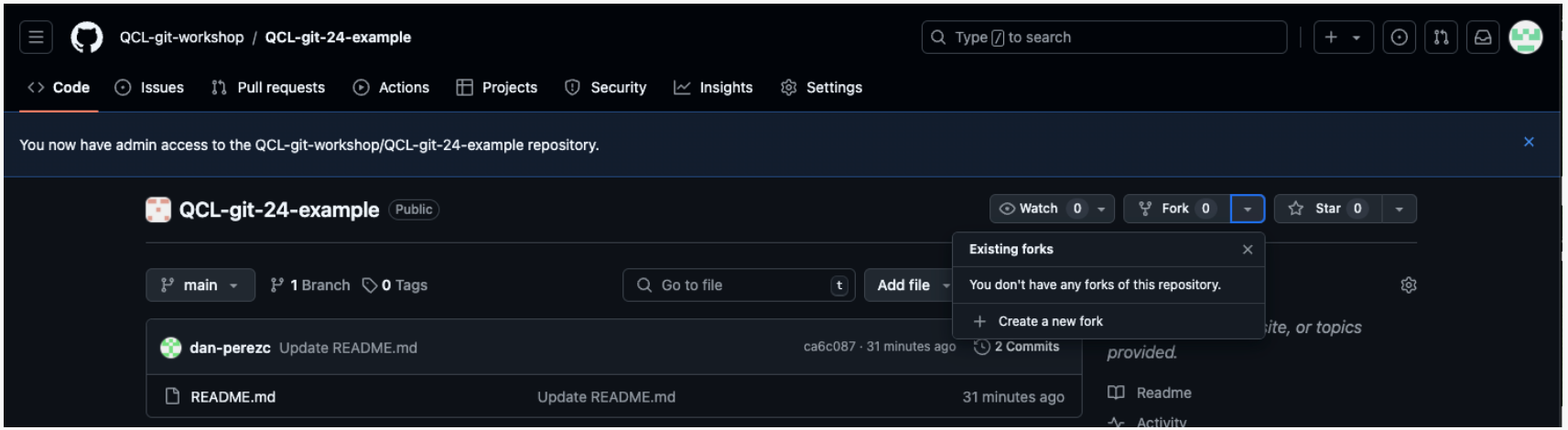


Prepare to work on a project



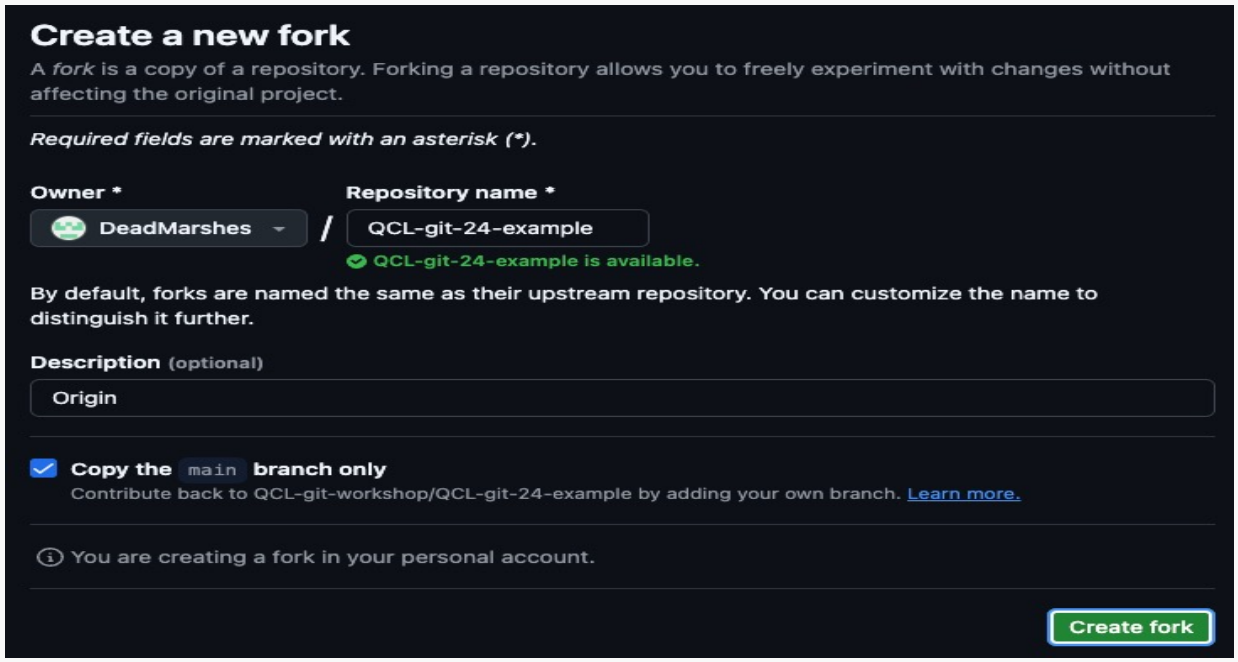
Activity! <Contributor>

Create new fork and clone the repo



Activity!

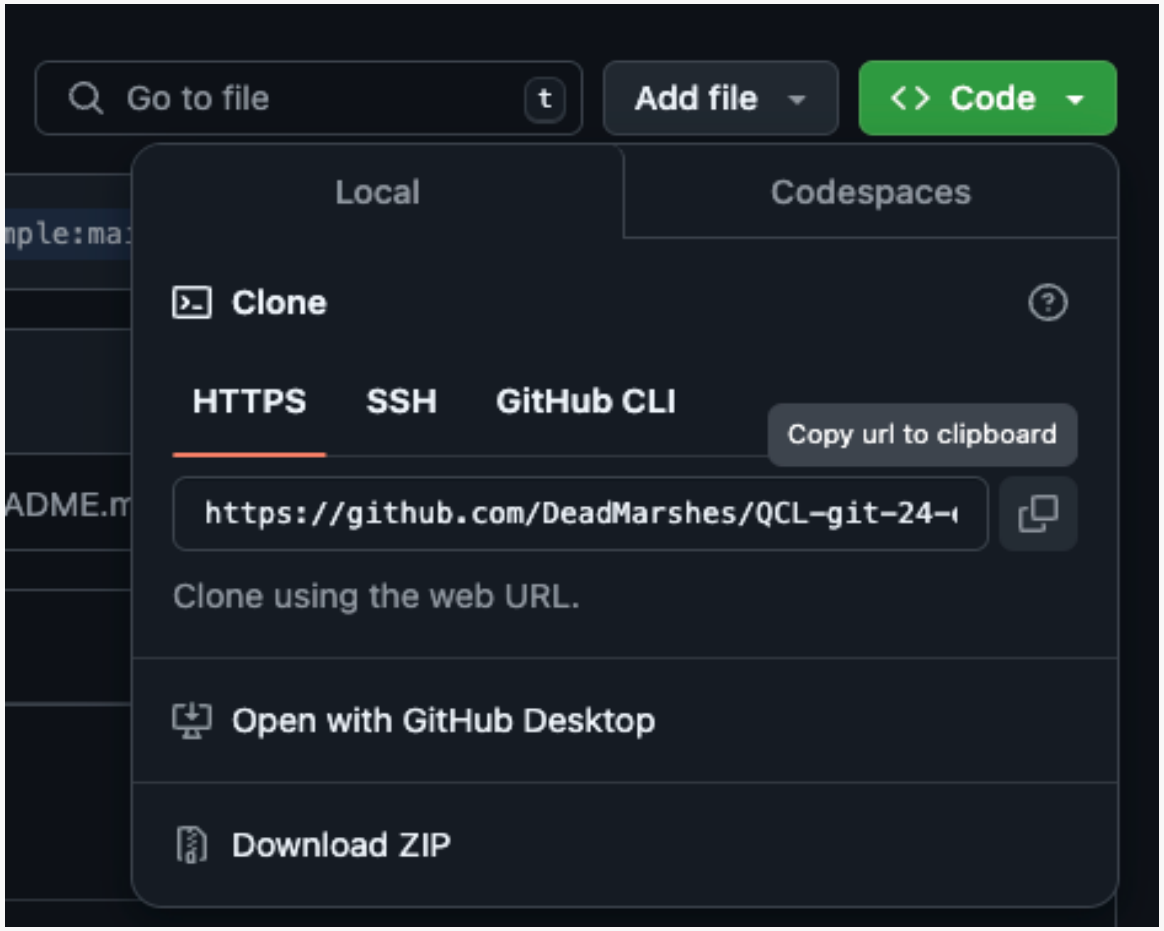
<Contributor>



Activity!

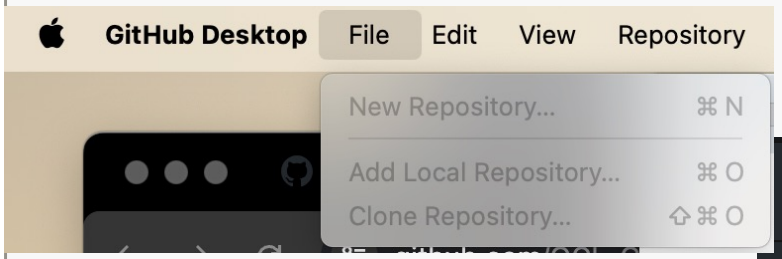
<Contributor>

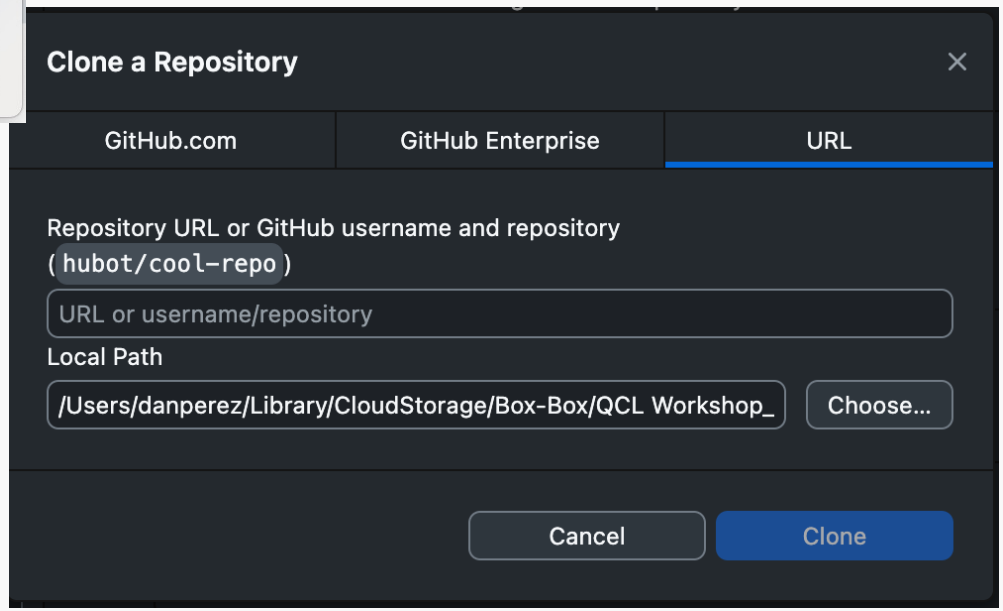
Make sure you are in your fork's page



Activity!

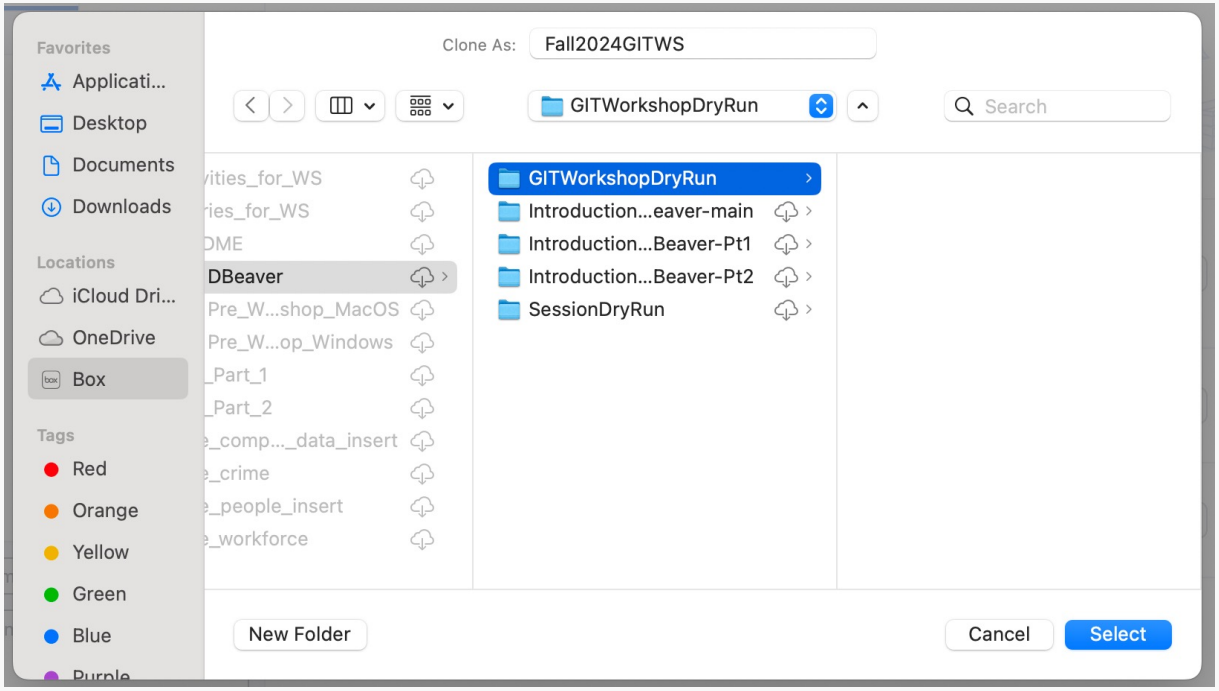
Clone your repo





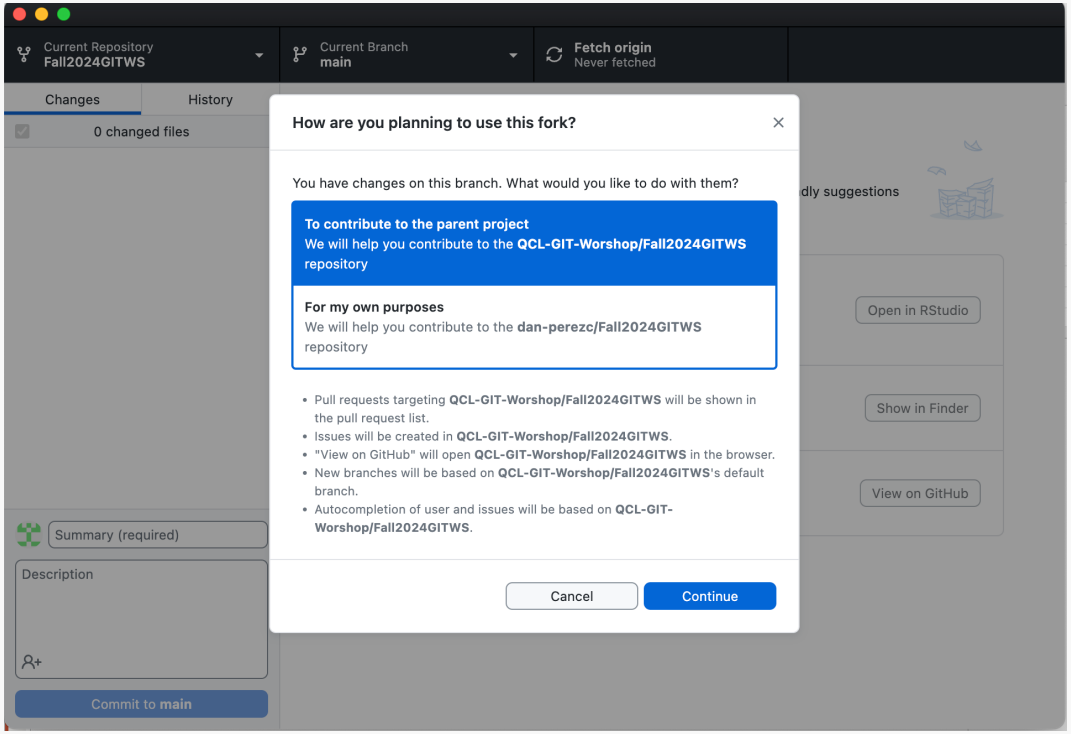
Activity!

Clone your repo



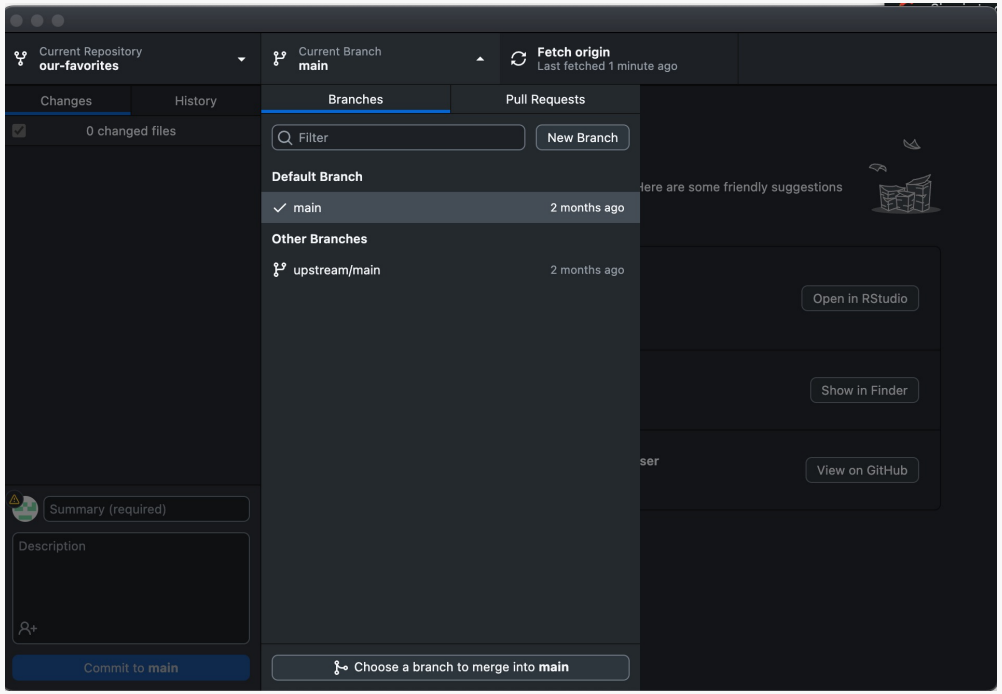
Activity!

Clone your repo



Activity!

Verify Branches, upstream remote

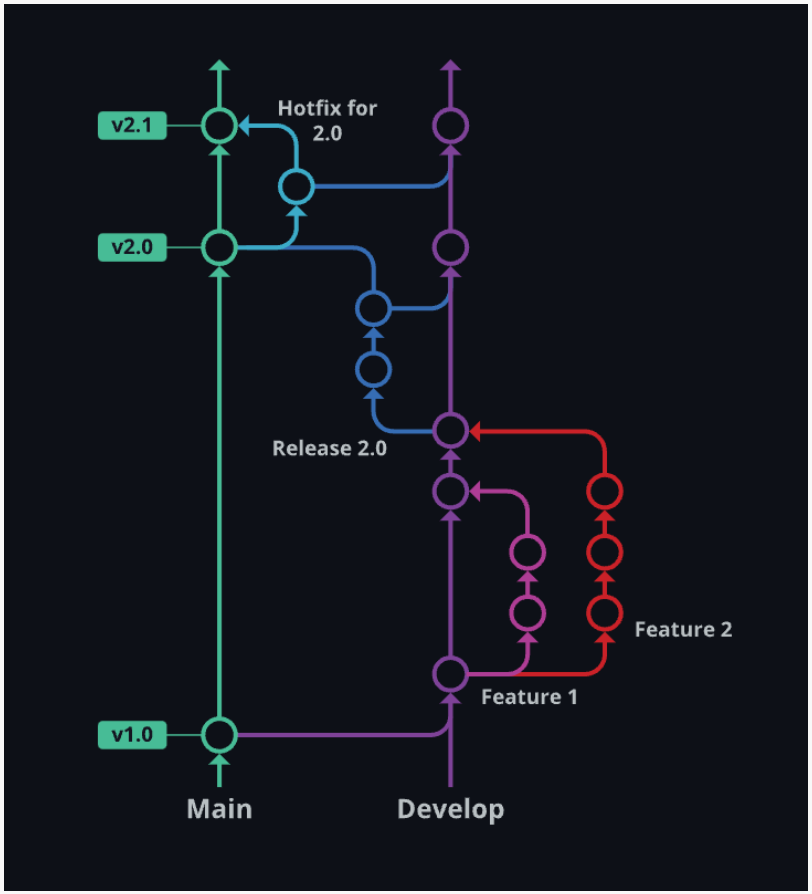


Branching Strategy

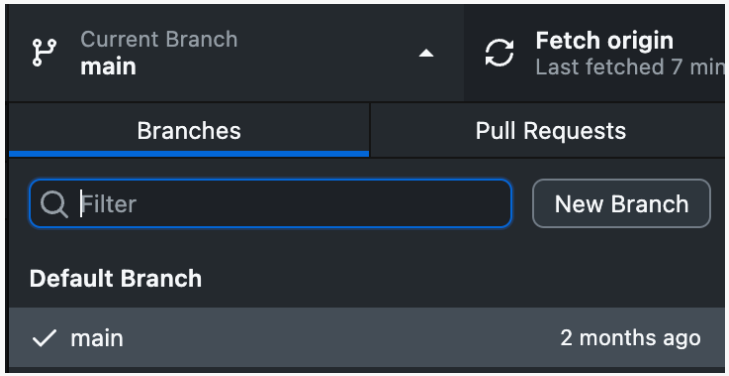
Branching allows developers to work on features or fixes in isolation without affecting the main codebase. It enables parallel development, experimentation, and the creation of feature branches.

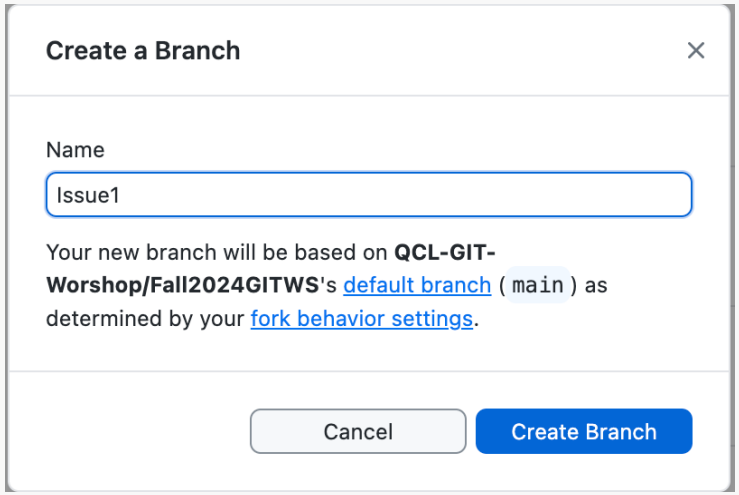
Best Practices

- Creating descriptive branch names, keeping branches short-lived
- merging changes frequently to avoid conflicts
- Having a branching strategy that aligns with the project's needs.

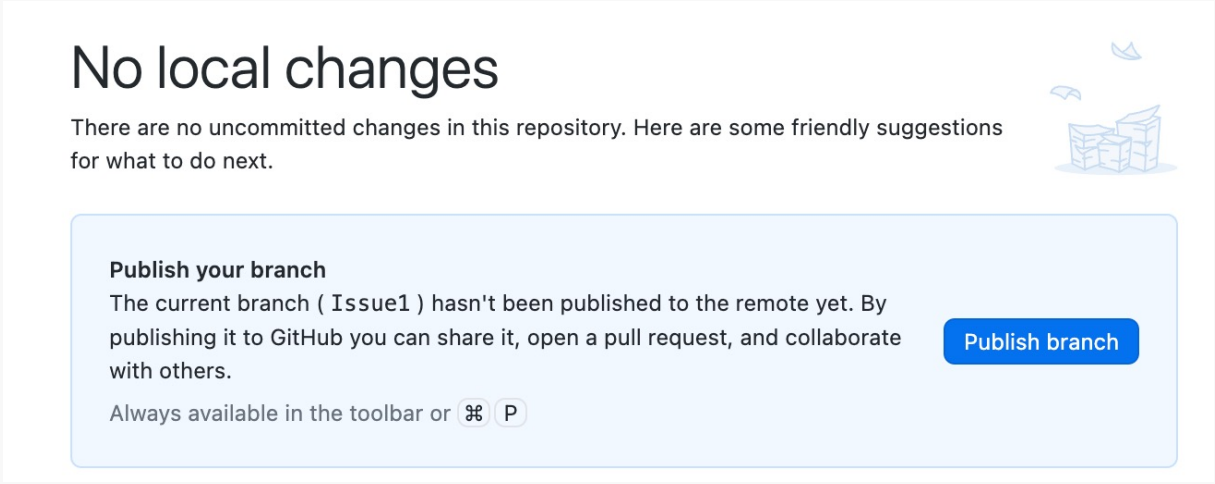


Check your assignment and Create a branch

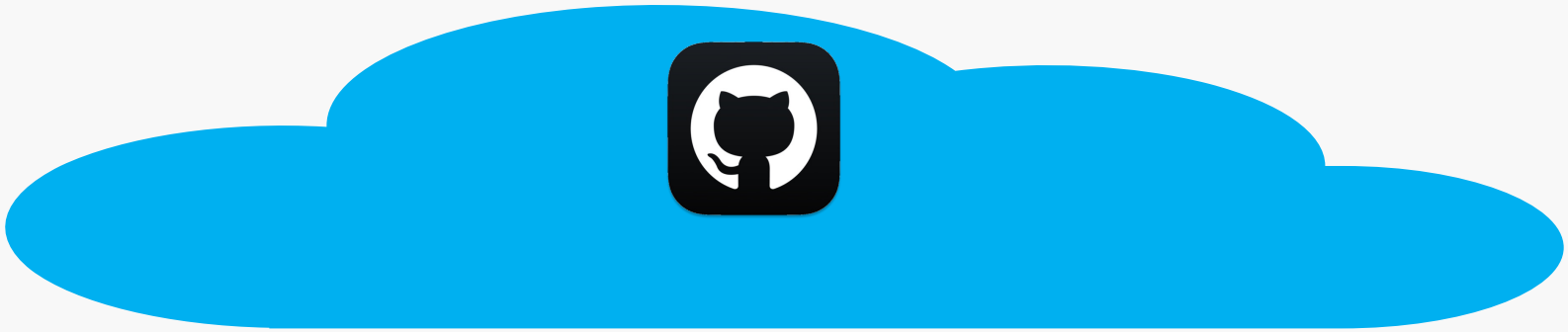




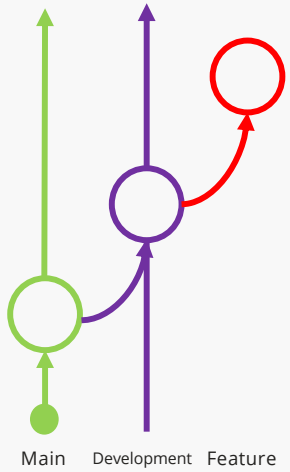
Create a branch



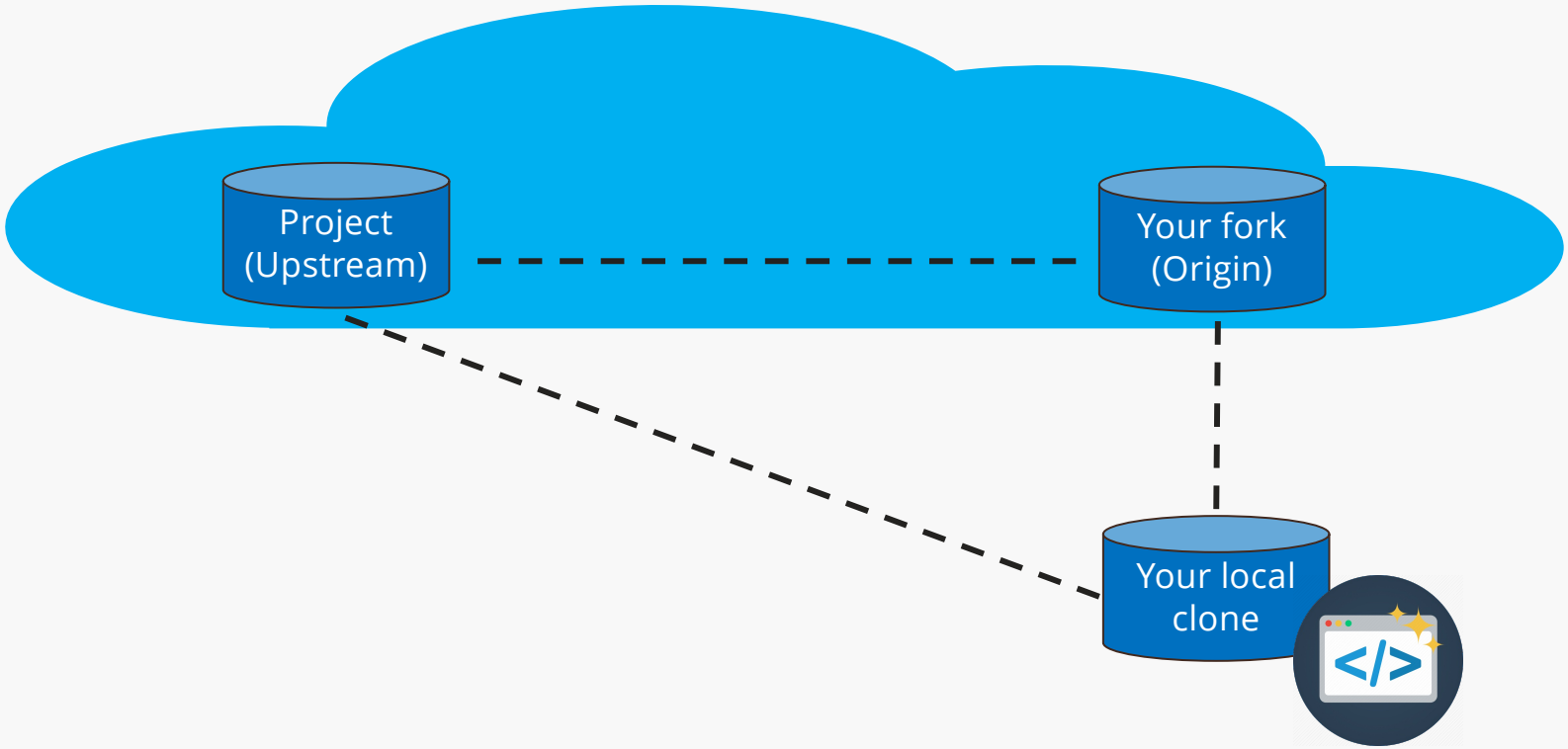
Project tracker



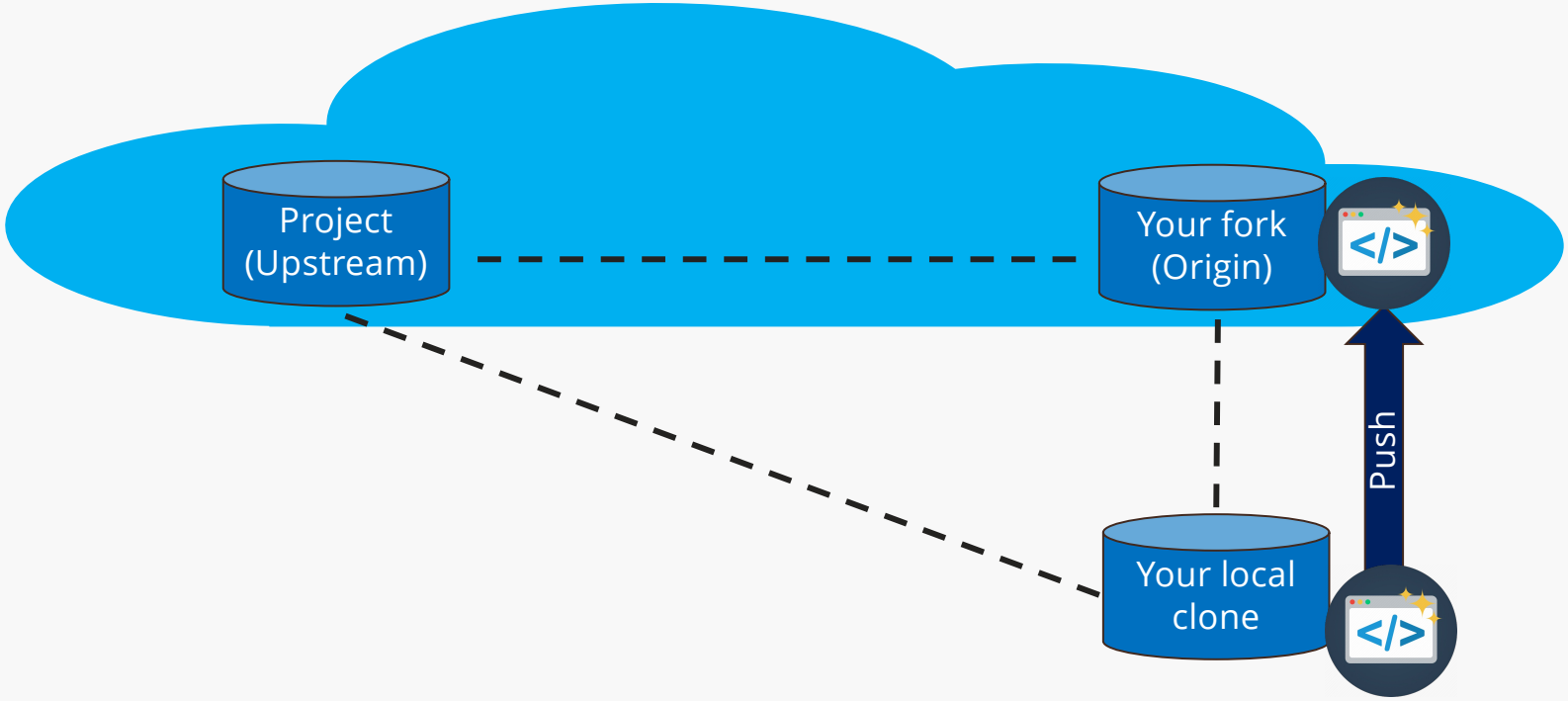
Branch/TAG	GRAPH	COMMIT	USER
Issue #1			Contributor
Development			Maintainer
Main/master		Update on file Readme	Maintainer



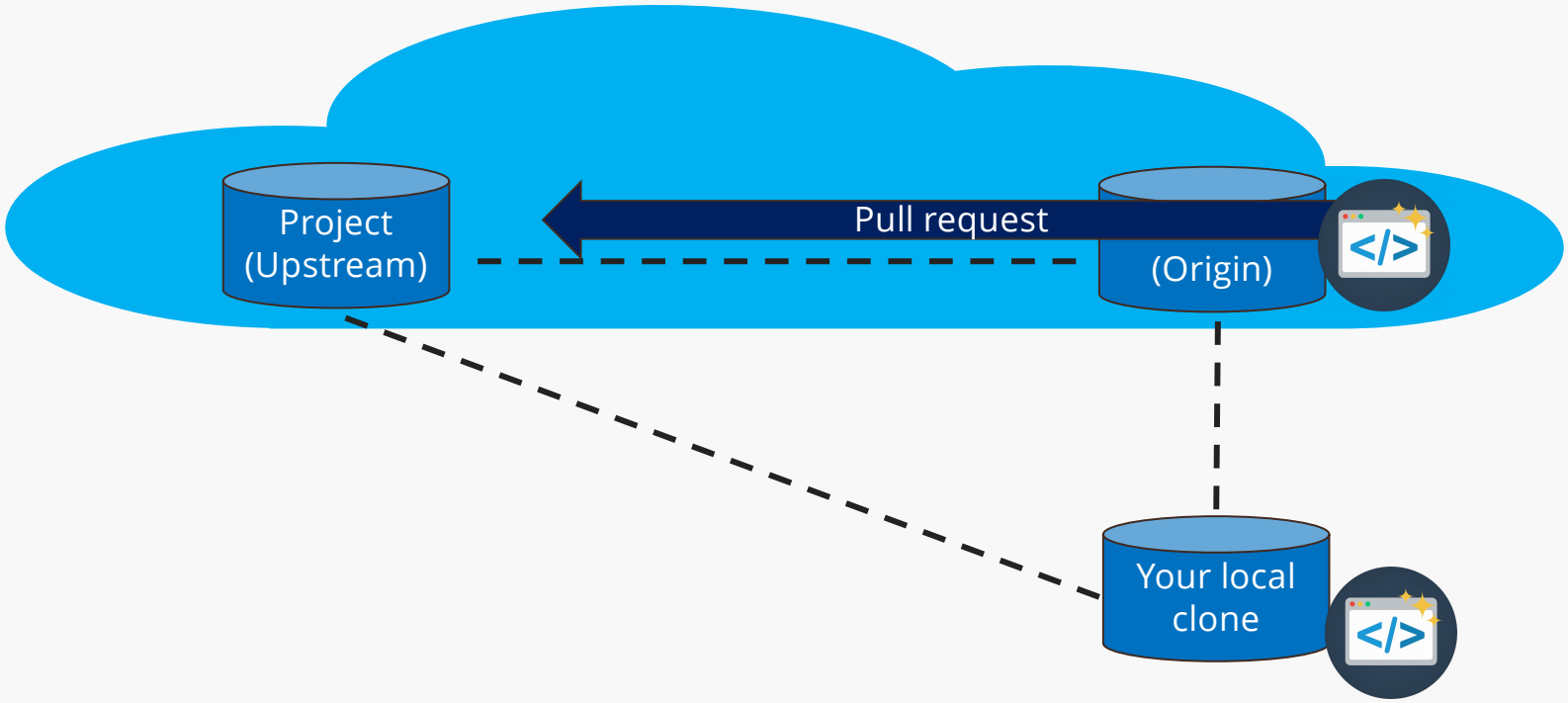
Contribute a change



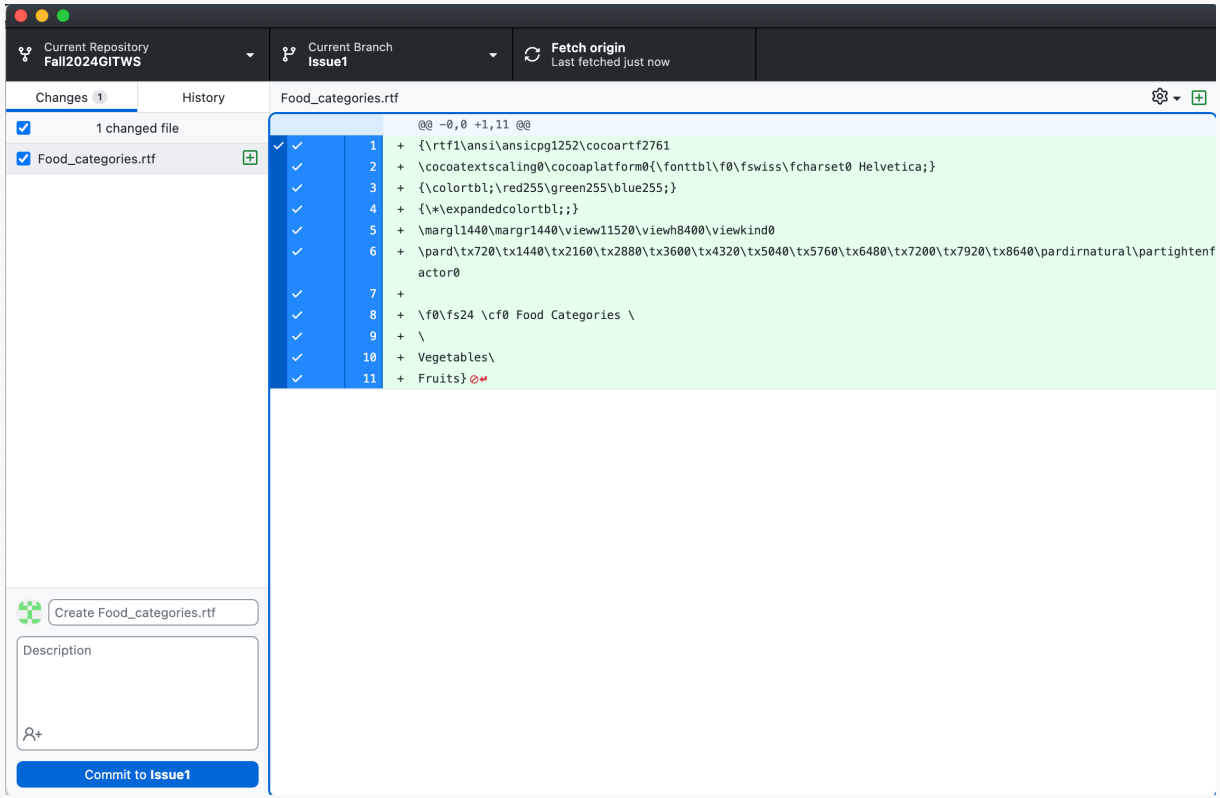
Contribute a change



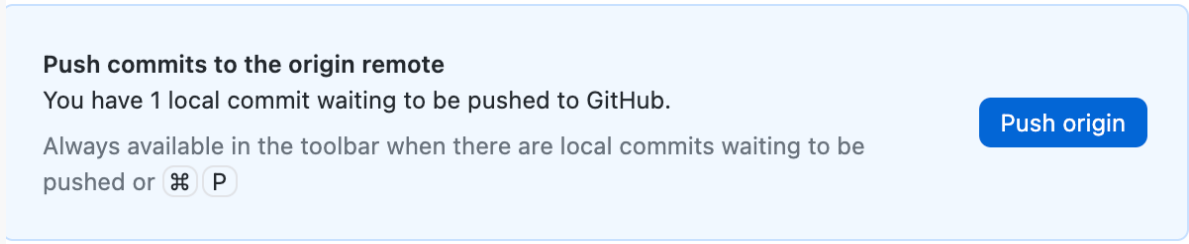
Contribute a change



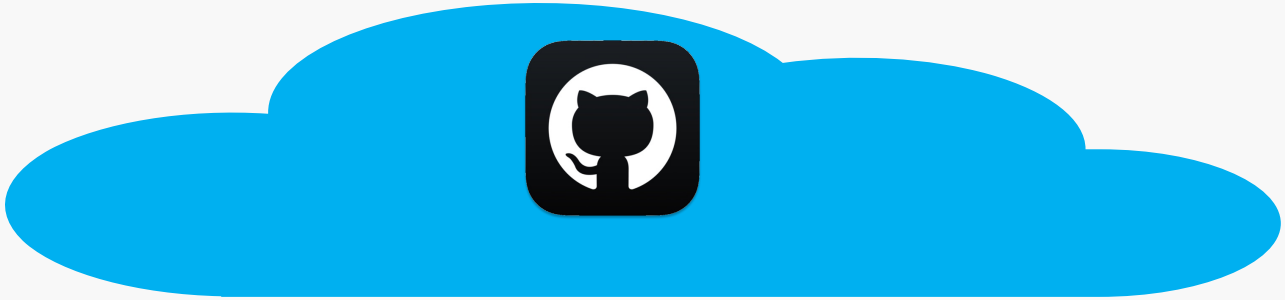
Activity! Work on your assigned issue

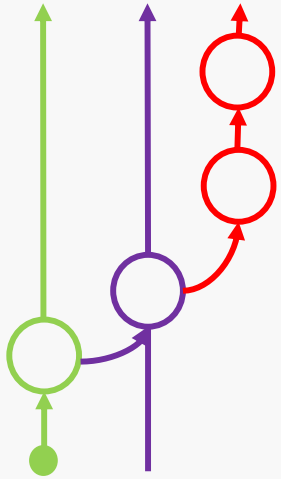


Activity! Push your changes to origin (Forked repo)



Project tracker

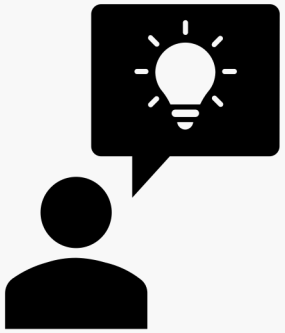


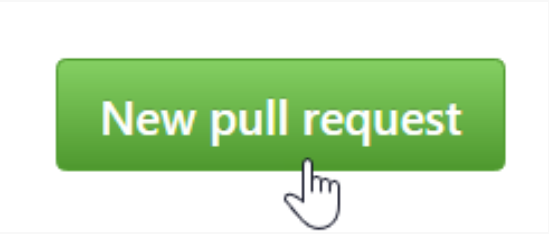
Branch/TAG	GRAPH	COMMIT	USER
Issue #1		Resolution Issue #1	Contributor
Issue #1			Contributor
Development			
Main/master		Update on file Readme	Maintainer

Pull Requests

Purpose of Pull Requests

Pull requests are used to propose changes, discuss modifications, and review code before merging it into the main branch. They allow for collaboration, feedback, and ensuring code quality.





Creation Process

When creating a pull request, developers provide context about the changes, request reviews from team members, run automated tests, and ensure that the code meets the project's standards before merging.

Collaboration and Code Review

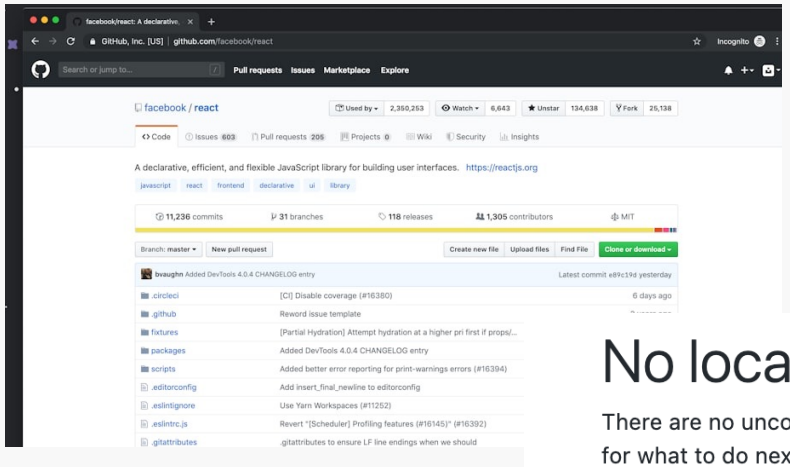
Facilitating Collaboration

GitHub facilitates collaboration by providing tools for code review, discussions, and pull request comments. It fosters teamwork, knowledge sharing, and error detection through code reviews.

Code Review Process

Code reviews in GitHub involve team members reviewing code changes, providing feedback, suggesting improvements, and ensuring code quality. They play a vital role in maintaining code consistency and quality standards.

Activity! Create Pull Requests



No local changes

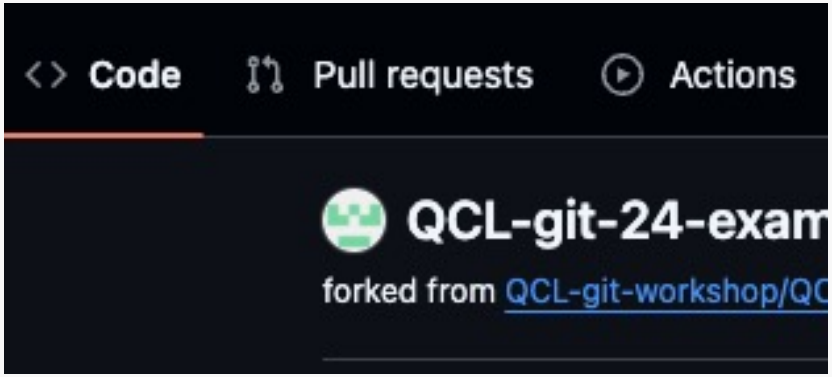
There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next.

Preview the Pull Request from your current branch
The current branch (`Issue1`) is already published to GitHub. Preview the changes this pull request will have before proposing your changes.
Branch menu or ⌘ ⌥ P

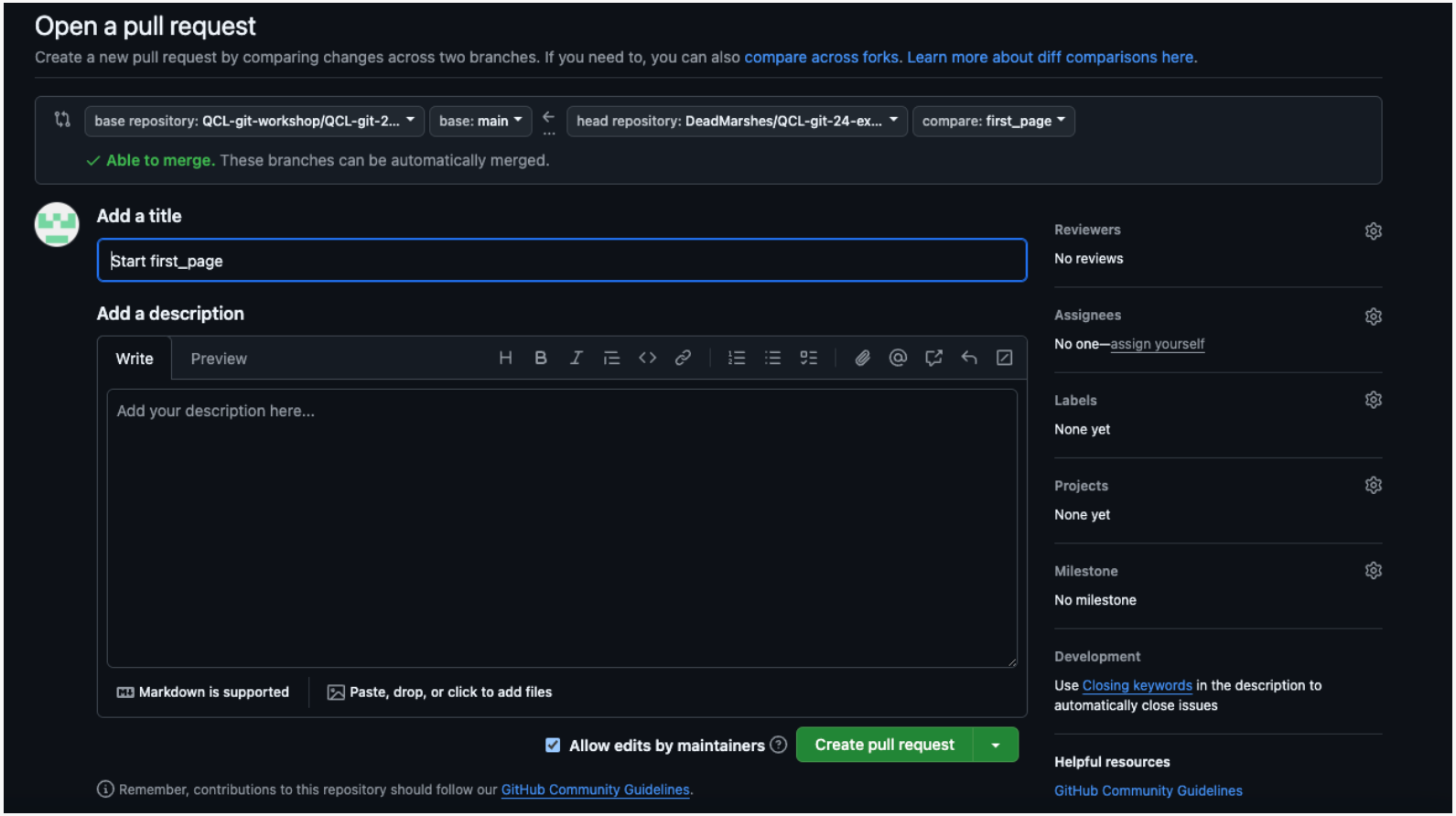
Preview Pull Request
✓ Preview Pull Request
Create Pull Request

Open the repository in your external editor

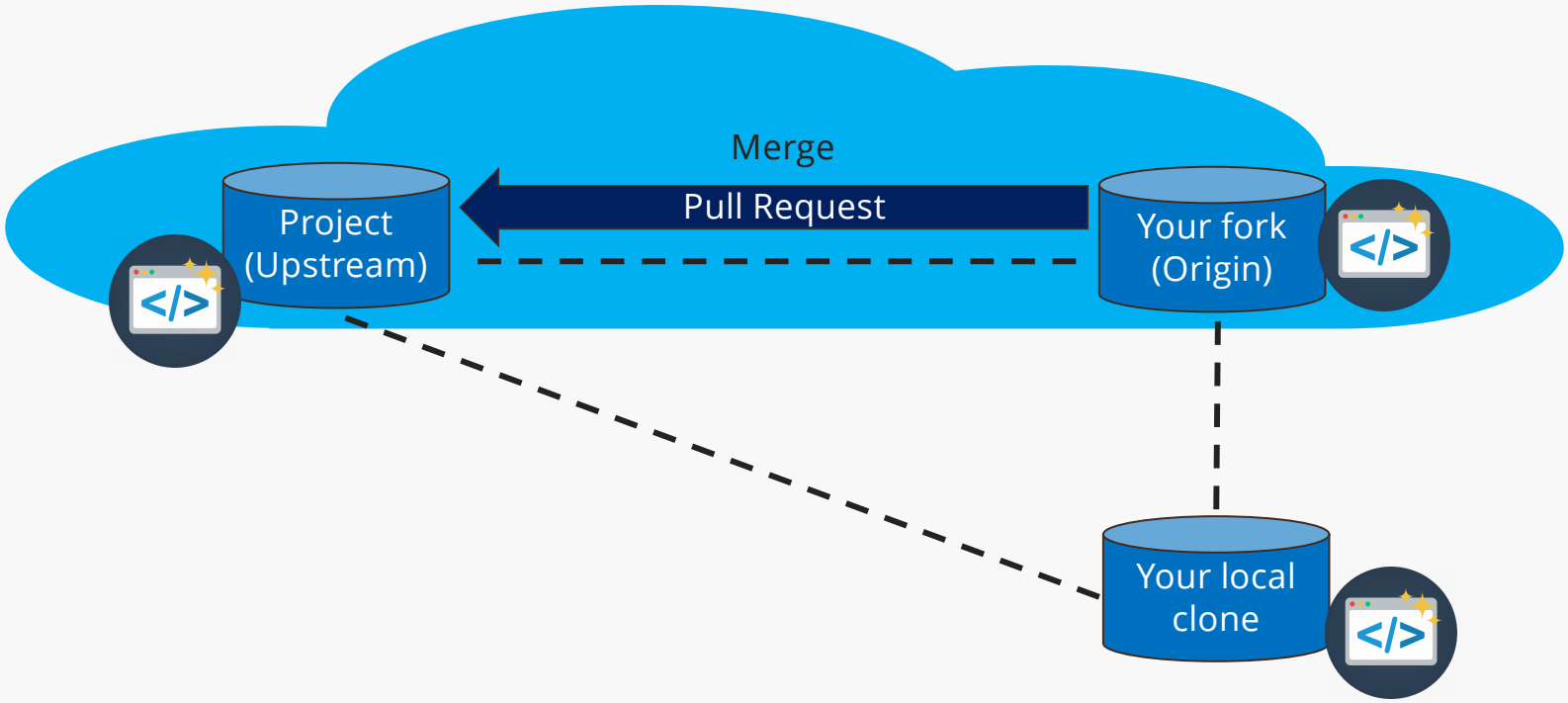
Activity! Pull Requests



Pull Requests



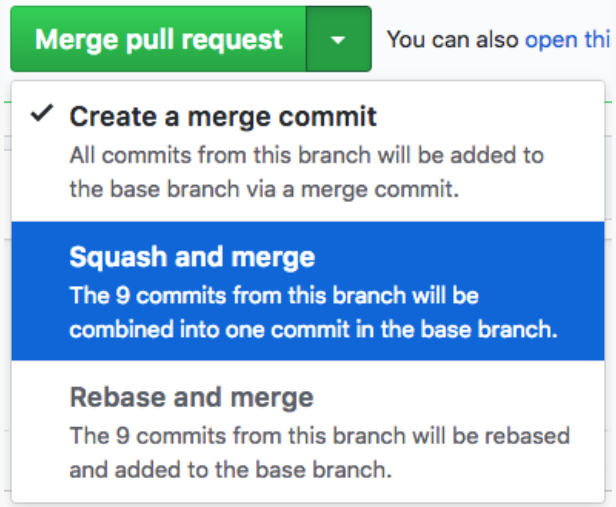
Integrate a change



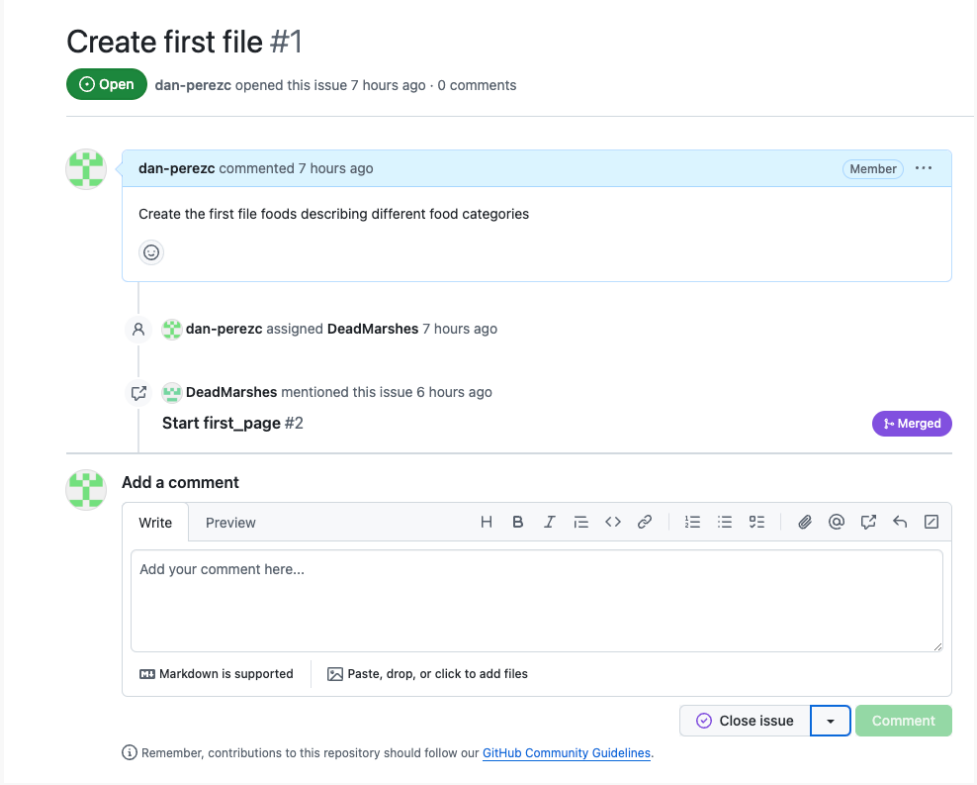
Maintainer reviews the PR

Looks good!

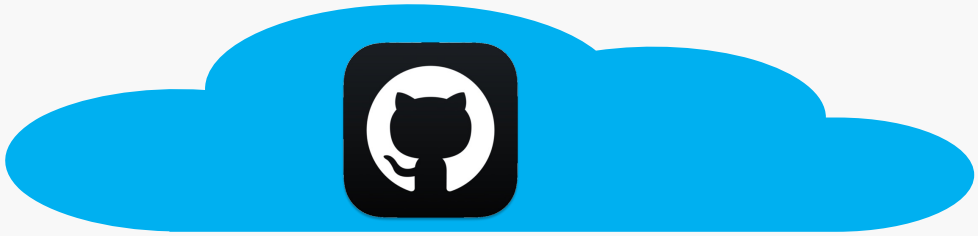
Accept the PR by merging it into master: choose "squash and merge" strategy for now and click **Merge pull request** and then **Confirm**.



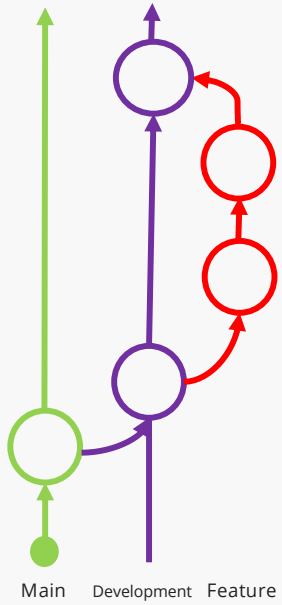
Maintainer close the issue



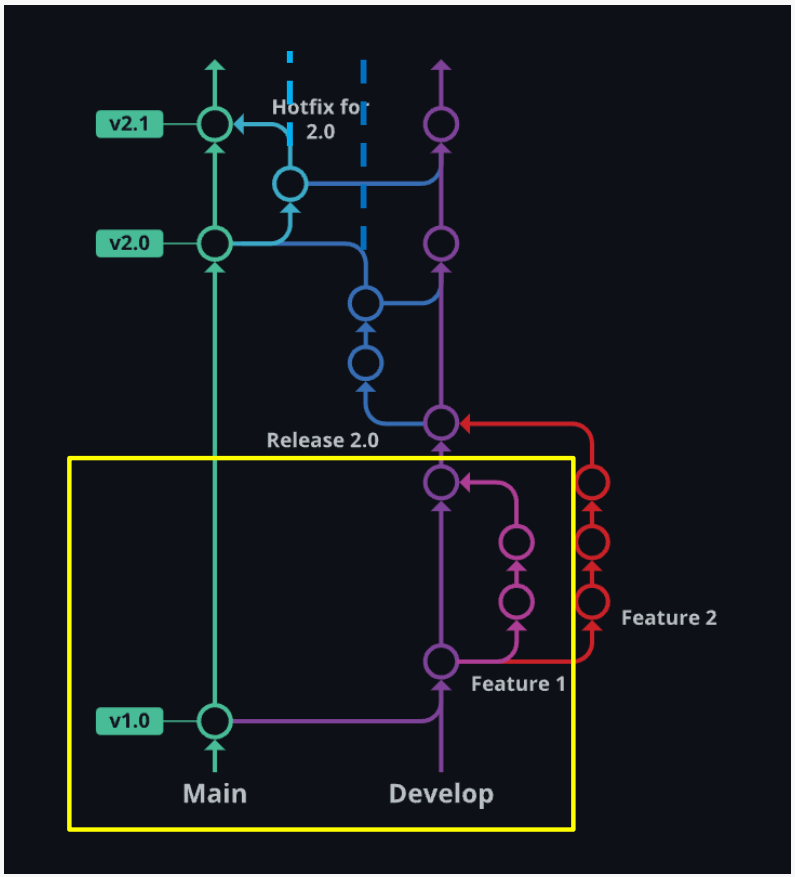
Project tracker



Branch/TAG	GRAPH	COMMIT	USER
Development		Merge Pull Request #1	Maintainer
Issue #1		Resolution Issue #1	Contributor
Issue #1		Work on issue #1	Contributor
Main/master		Update on file Readme	Maintainer



Branching Strategy



Let’s cleanup!

Stand on main branch and pull upstream

Delete feature branch and push the change to origin

Workflow Best Practices

Best Practice	Description
Branch Protection	Enable branch protection rules to prevent direct commits to critical branches and ensure changes go through code review.
Continuous Integration	Integrate CI tools to automatically test and build code changes, ensuring the project's stability and quality.
Documentation	Maintain detailed documentation in repositories to enhance project accessibility, onboarding, and knowledge sharing.
Code Reviews	Encourage regular code reviews to catch bugs early, share knowledge, and maintain code quality standards.
Issue Tracking	Effectively use GitHub issues to track tasks, bugs, and enhancements, facilitating project management and collaboration.

Additional Info

https://docs.github.com/en/pull-requests/collaborating-with-pull-requests/proposing-changes-to-your-work-with-pull-requests/creating-and-deleting-branches-within-your-repository

https://www.geeksforgeeks.org/version-control-systems/

https://www.geeksforgeeks.org/ultimate-guide-git-github/?ref=gcse_ind

https://www.gitkraken.com/learn/git/git-flow

https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow