Data Science in Bioinformatics

WS 22/23

Bioinformatics basics and tooling 24.10.2022

High level overview

- This session: Tips and tricks from experience
- The goal: Lay a common groundwork to base the rest of the course on
- Topics we will cover:
  - git basics
  - Conda / Mamba / environments and package managers
  - Software development setups or “How do I get productive on the cluster”
  - Containers and VMs

Git basics

- Distributed source code version control software
- Build in 2005 by Linus Torvalds for the Linux kernel
- Very powerful and widely used around the globe
- Mainly build for source code, not large binary files (there are extensions for this)
- Not to be confused with „GitHub“ and „GitLab“.

Git basics

- Most important commands and what they do:

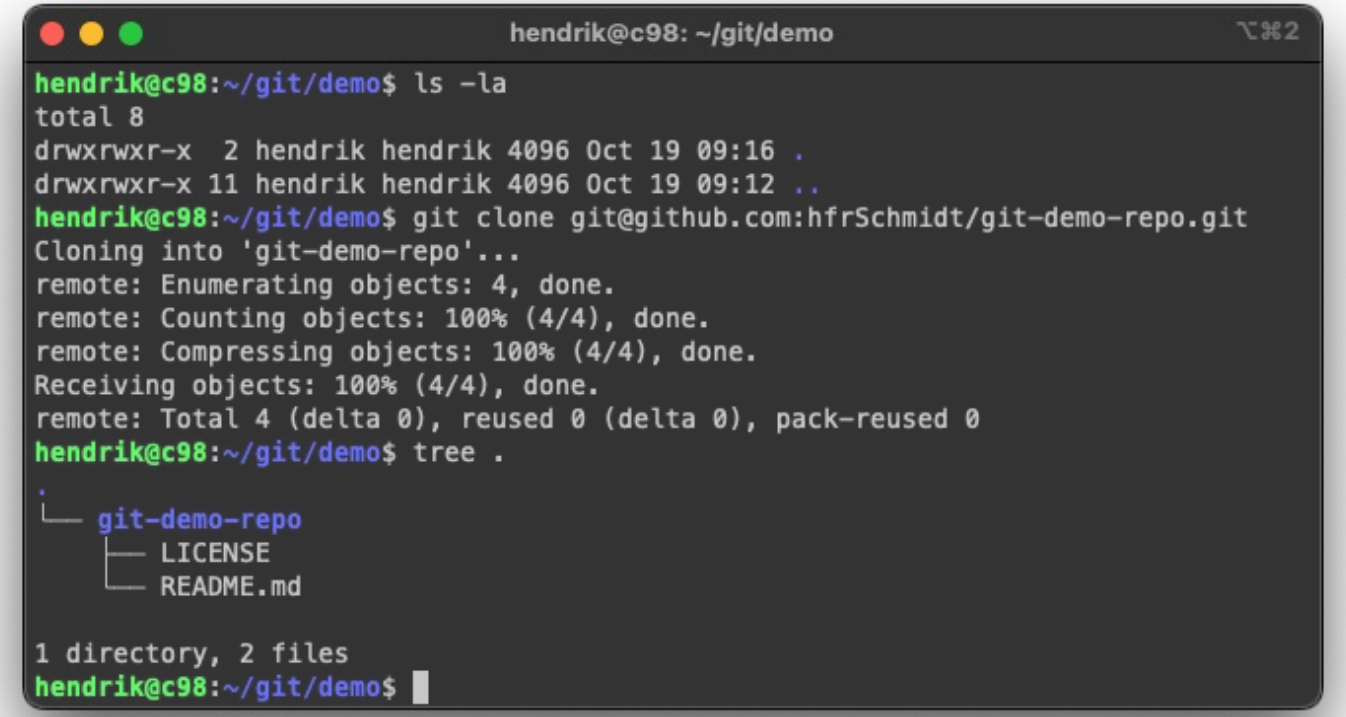
git clone	Initially copy new repository to your local machine
git pull	Update the current branch of your local copy from remote
git add	Add a file or directory to the staging area
git commit	Sum up all changes from the staging area to a change / commit
git push	Upload local commits to remote repository
git branch –a	Show all references to branches (local + remote)
git checkout	Open a new/existing branch

Git as a service providers

- Github vs Gitlab vs Bitbucket vs …
  - This course: Github
- Not to be confused with git itself
- Issues + Pull requests
- Continuous integration / Continuous delivery (CI/CD)
  - Automatic builds / tests
  - Automatic deployment to staging / production systems
  - Not necessarily connected to git

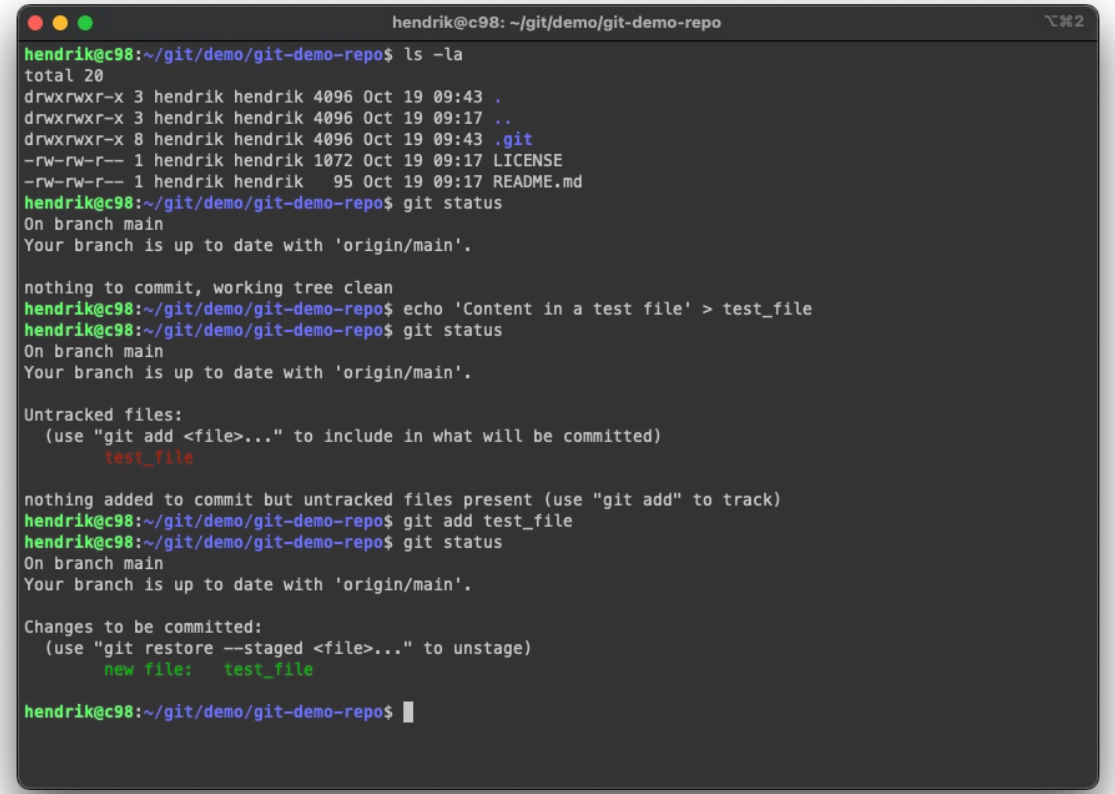
Git in action

- Clone a repository



Git in action

- Add a file to the repo



Git in action

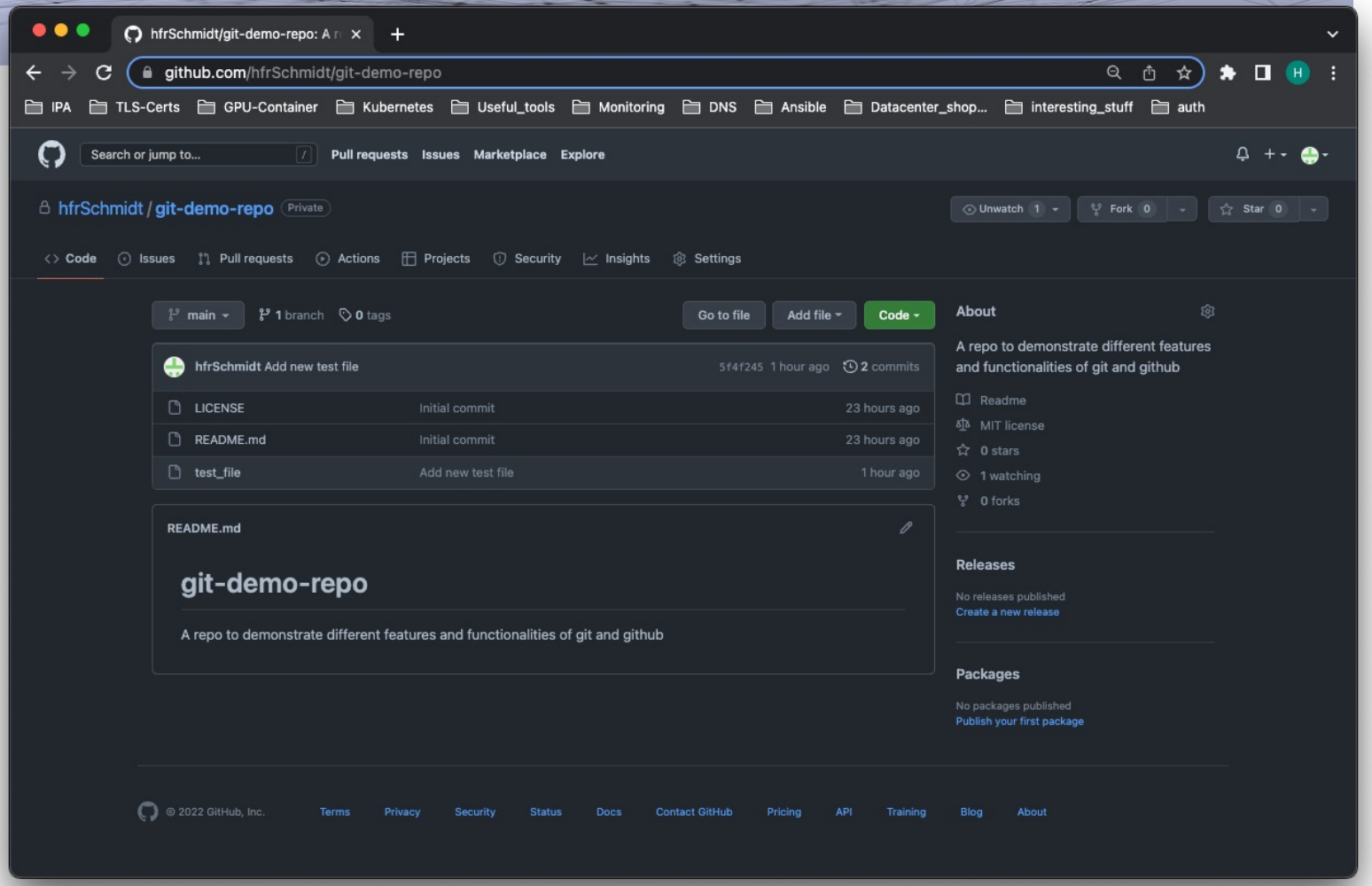
- Commit & push the file to the remote repo

```
hendrik@c98: ~/git/demo/git-demo-repo
hendrik@c98:~/git/demo/git-demo-repo$ git commit -v -m 'Add new test file'
[main 5f4f245] Add new test file
 1 file changed, 1 insertion(+)
 create mode 100644 test_file
hendrik@c98:~/git/demo/git-demo-repo$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
hendrik@c98:~/git/demo/git-demo-repo$
```

```
hendrik@c98: ~/git/demo/git-demo-repo
hendrik@c98:~/git/demo/git-demo-repo$ git push
git@ssh.github.com: Permission denied (publickey).
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.
hendrik@c98:~/git/demo/git-demo-repo$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 64 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 351 bytes | 351.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:hfrSchmidt/git-demo-repo.git
   4169393..5f4f245  main -> main
hendrik@c98:~/git/demo/git-demo-repo$
```

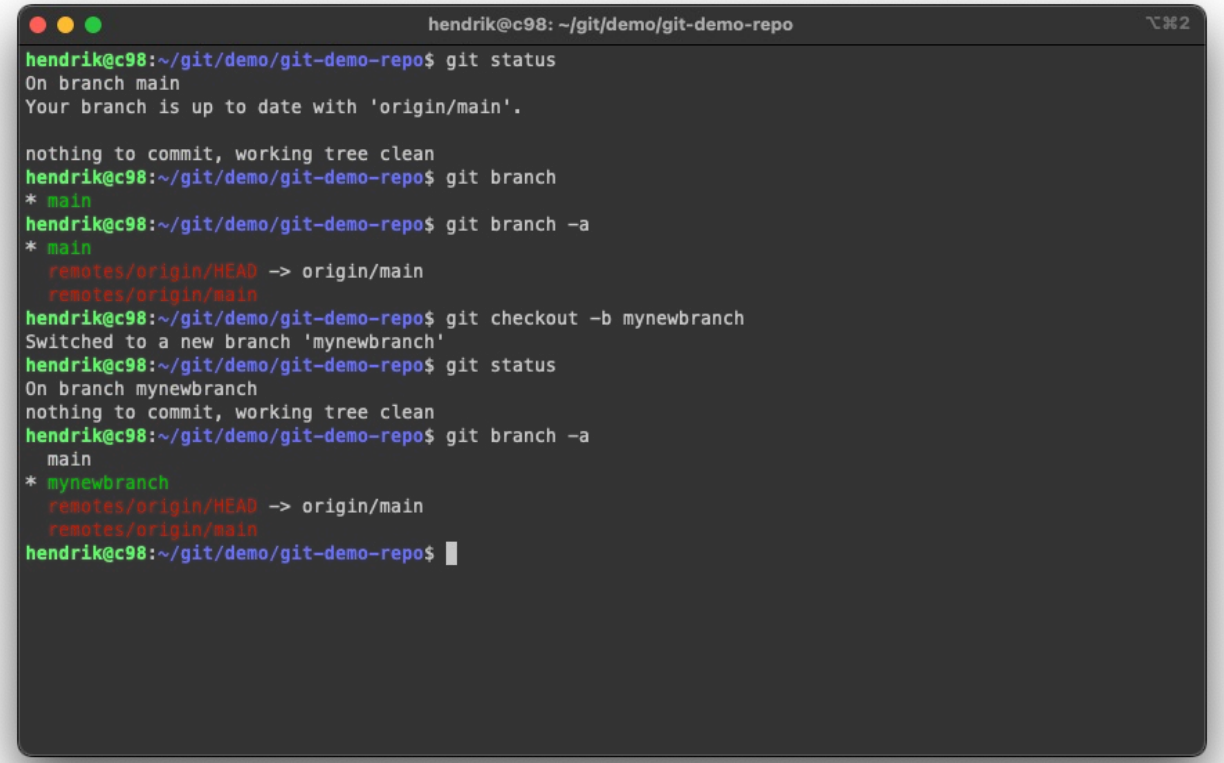
Git in action

- View from the remote side of the repo
- Currently: working on the ”main” branch



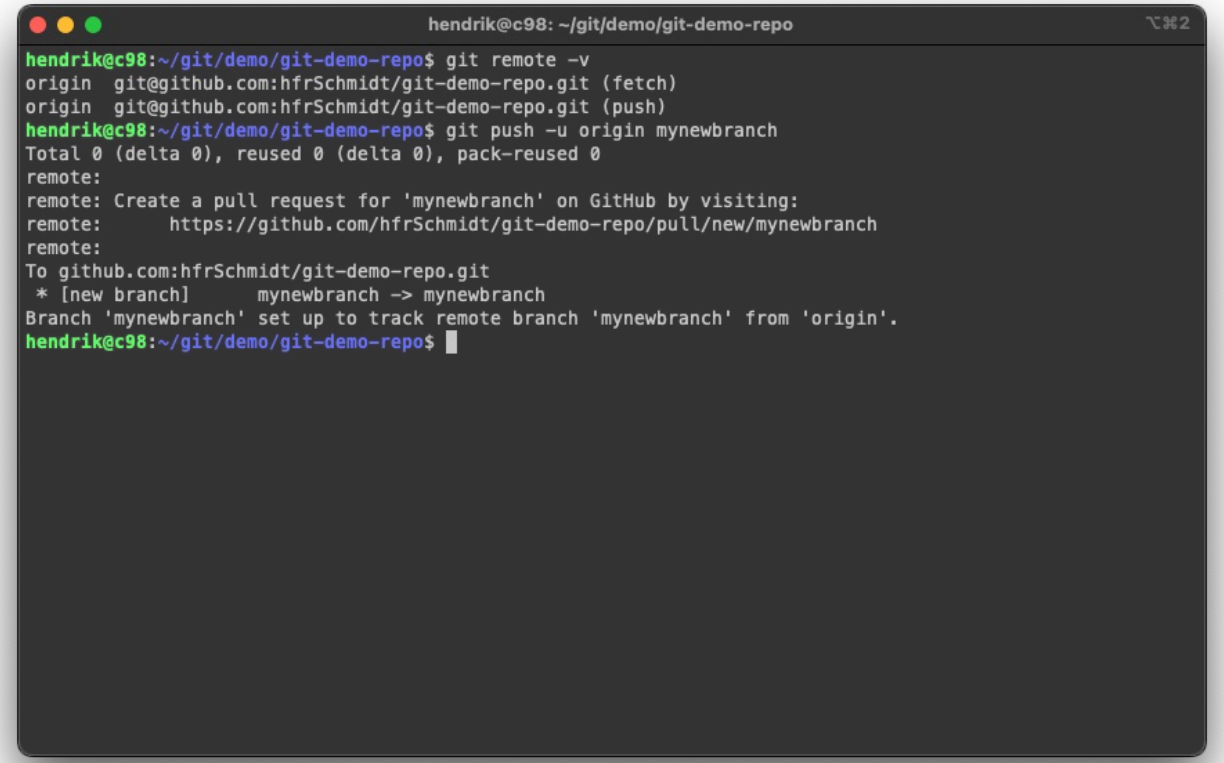
Git in action

- Create a new branch



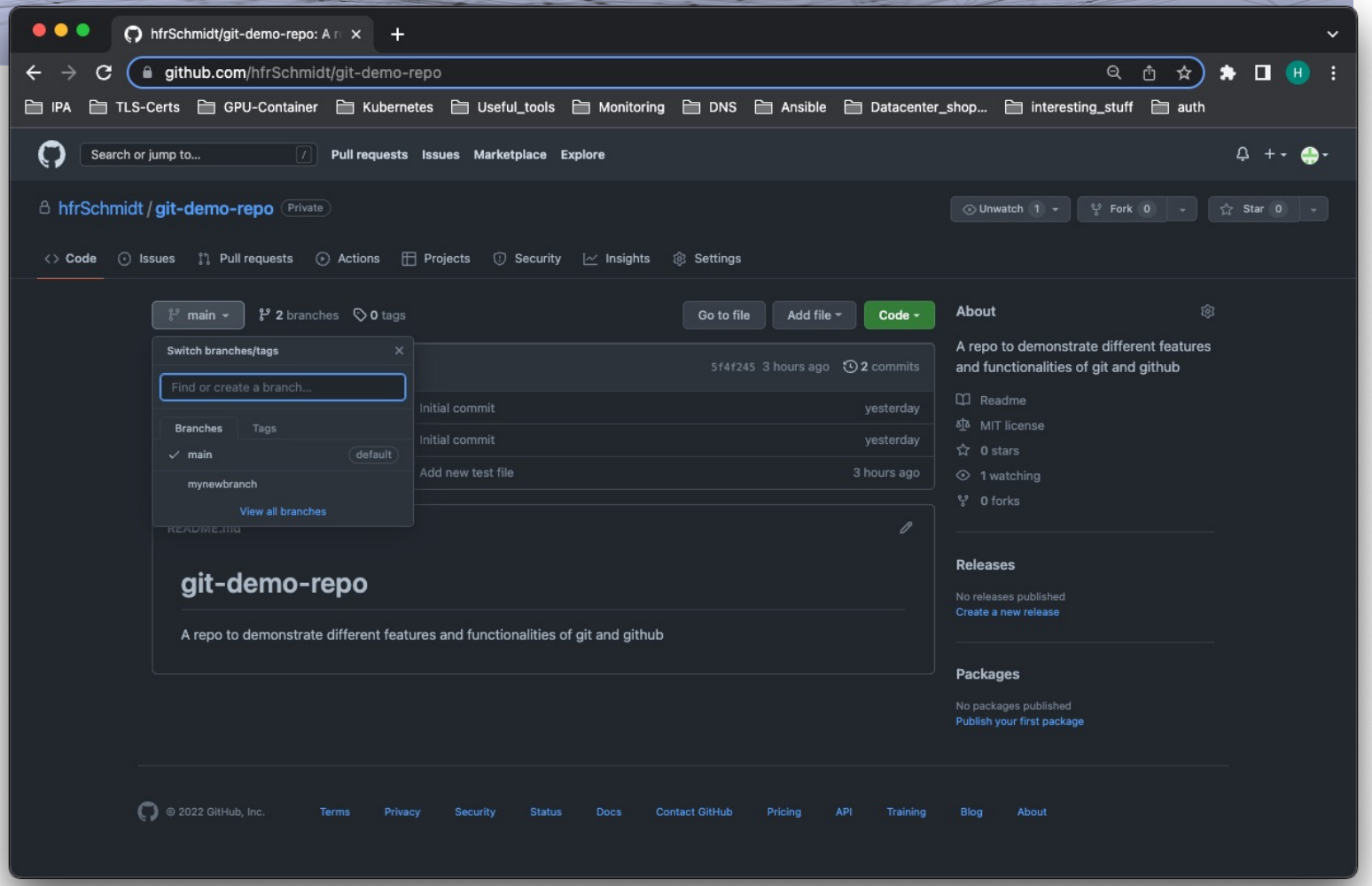
Git in action

- Push the new branch to the remote



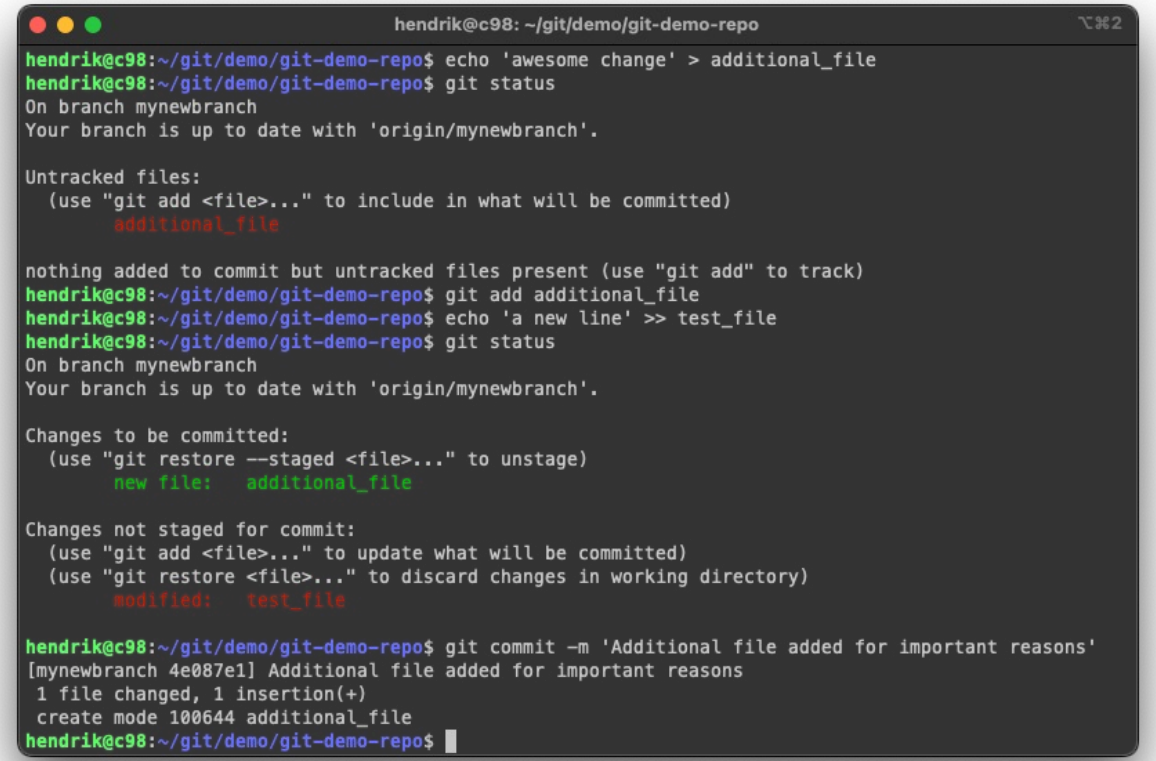
Git in action

- View from the remote side of the repo
- The new branch is there



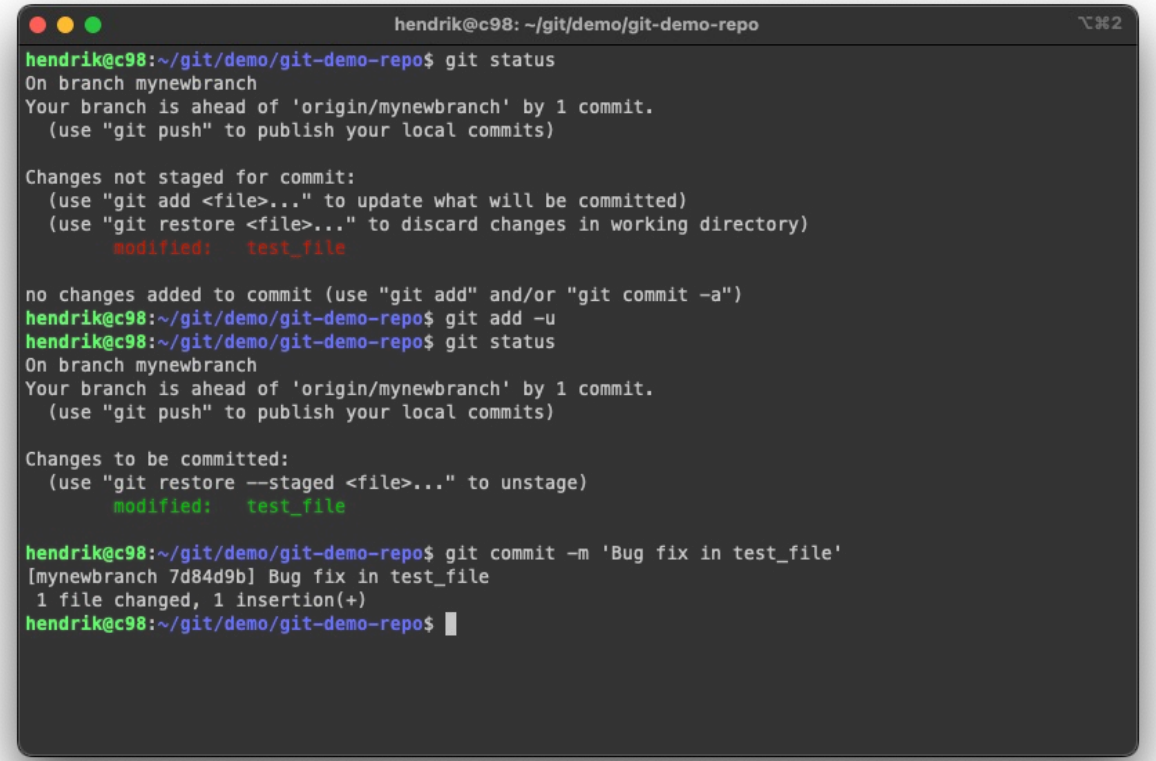
Git in action

- Do some work on the new branch



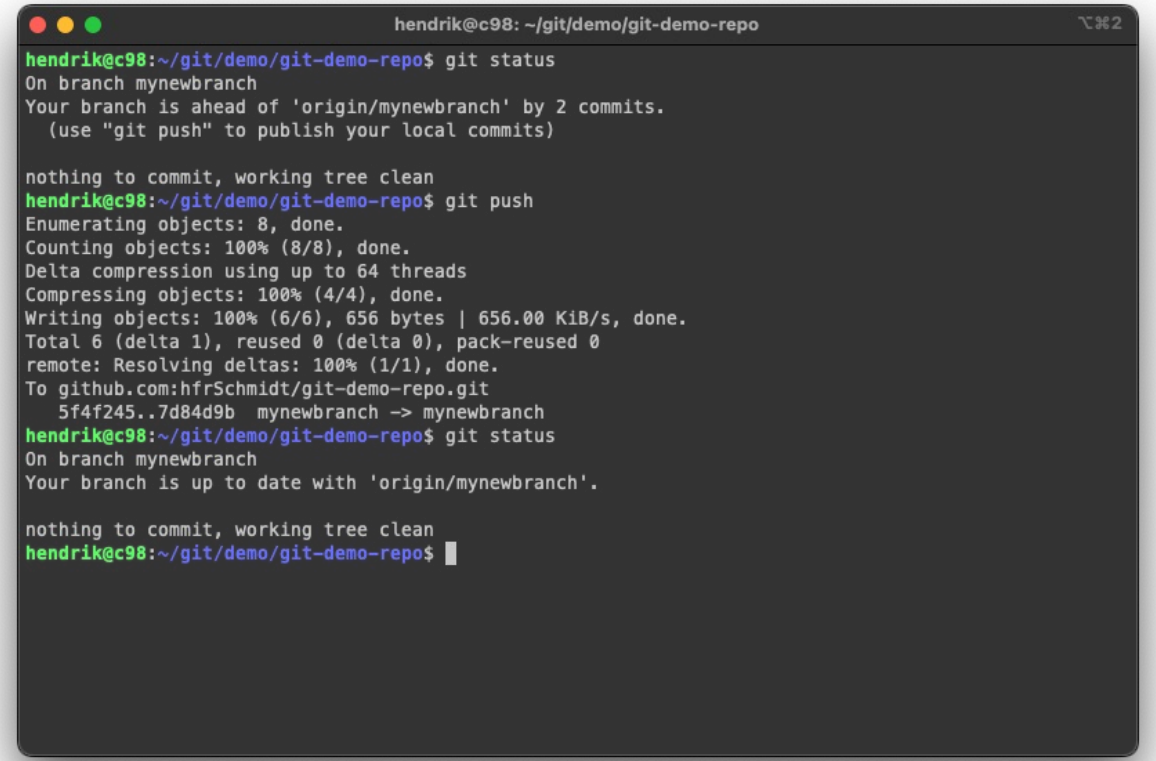
Git in action

- Do some more work on the new branch



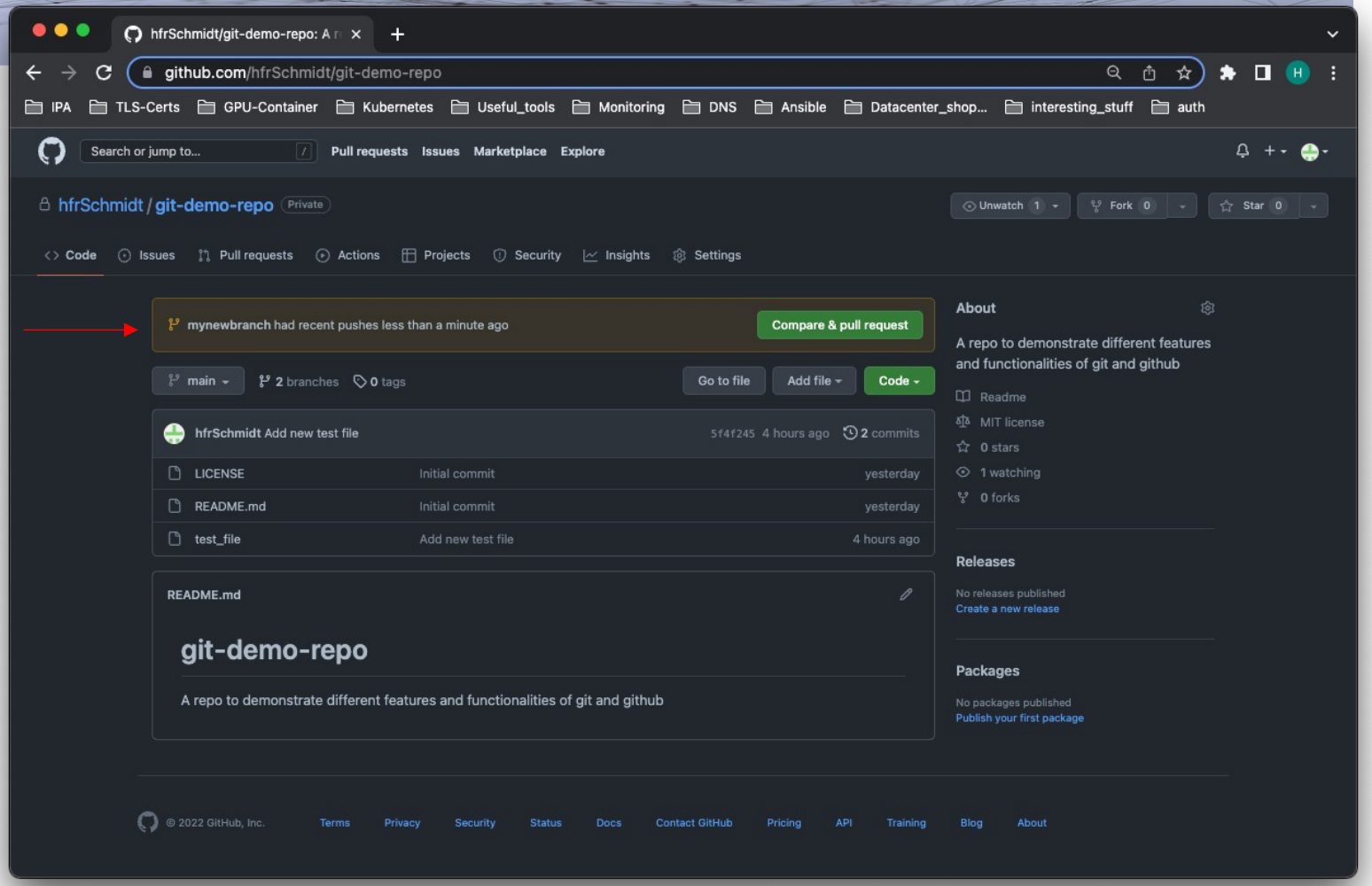
Git in action

- Push the changes to the remote branch



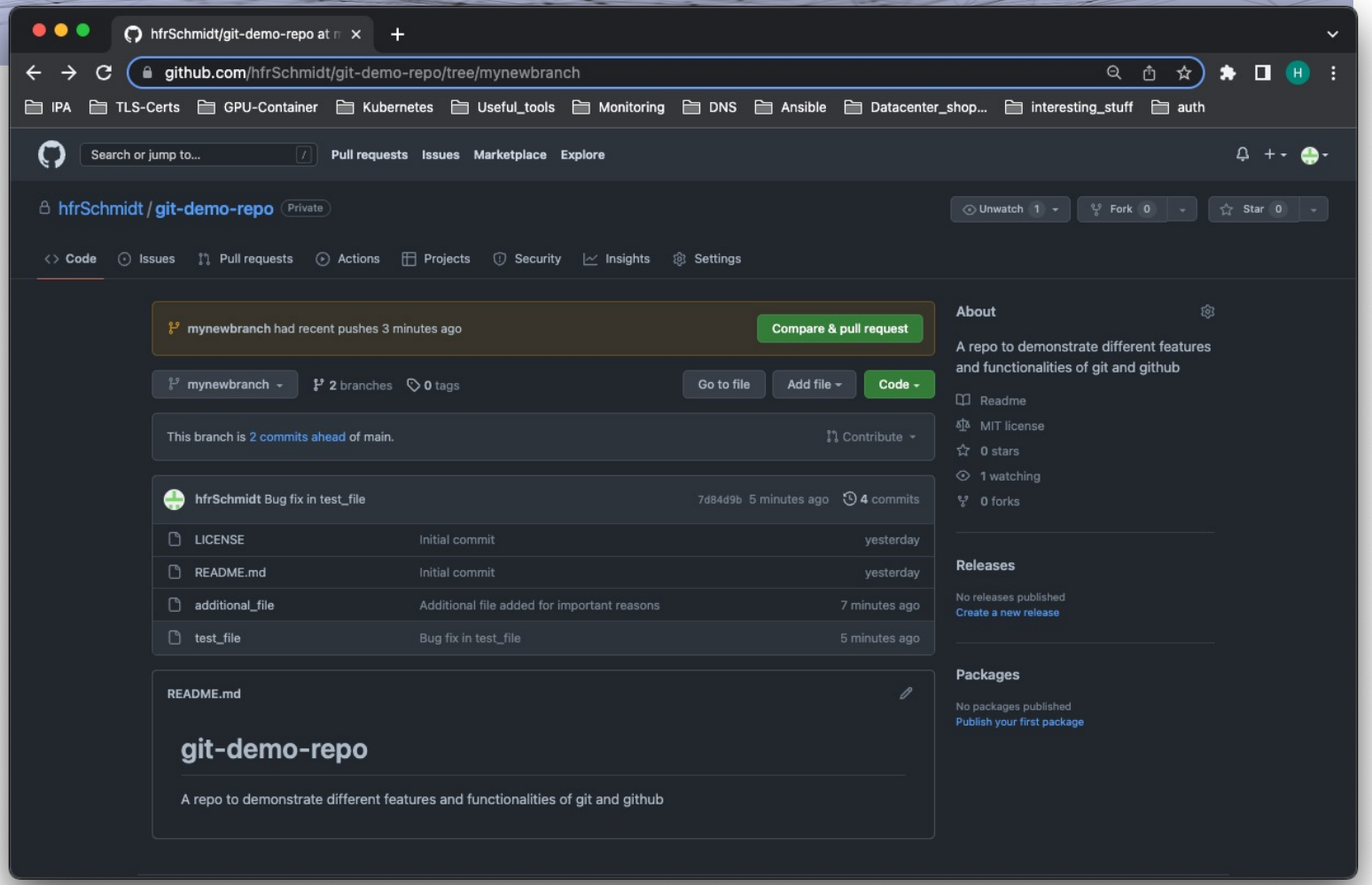
Git in action

- View from the remote side of the repo
- Changes not visible in “main”



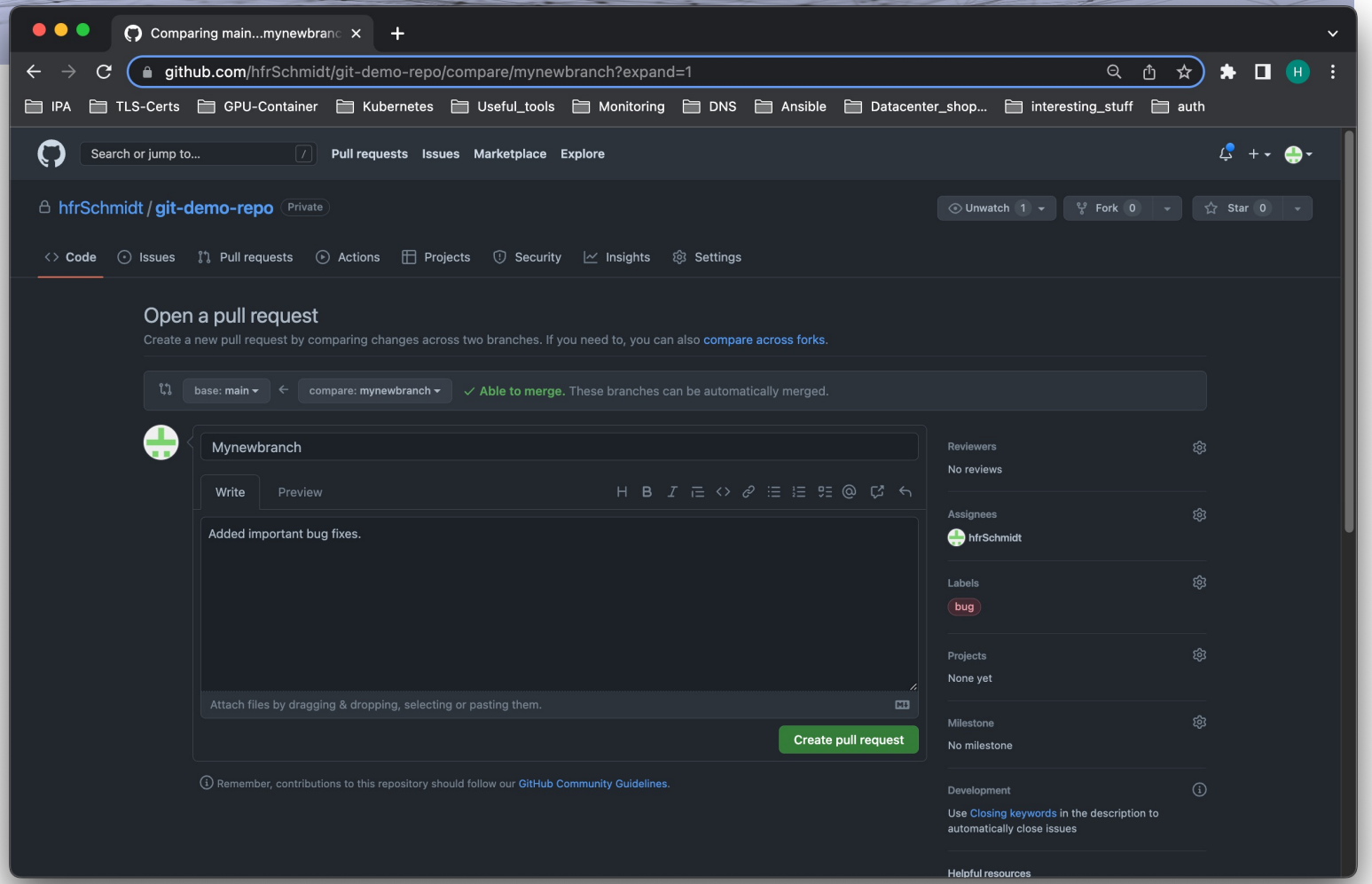
Git in action

- View from the remote side of the repo
- ”mynewbranch” has the changes



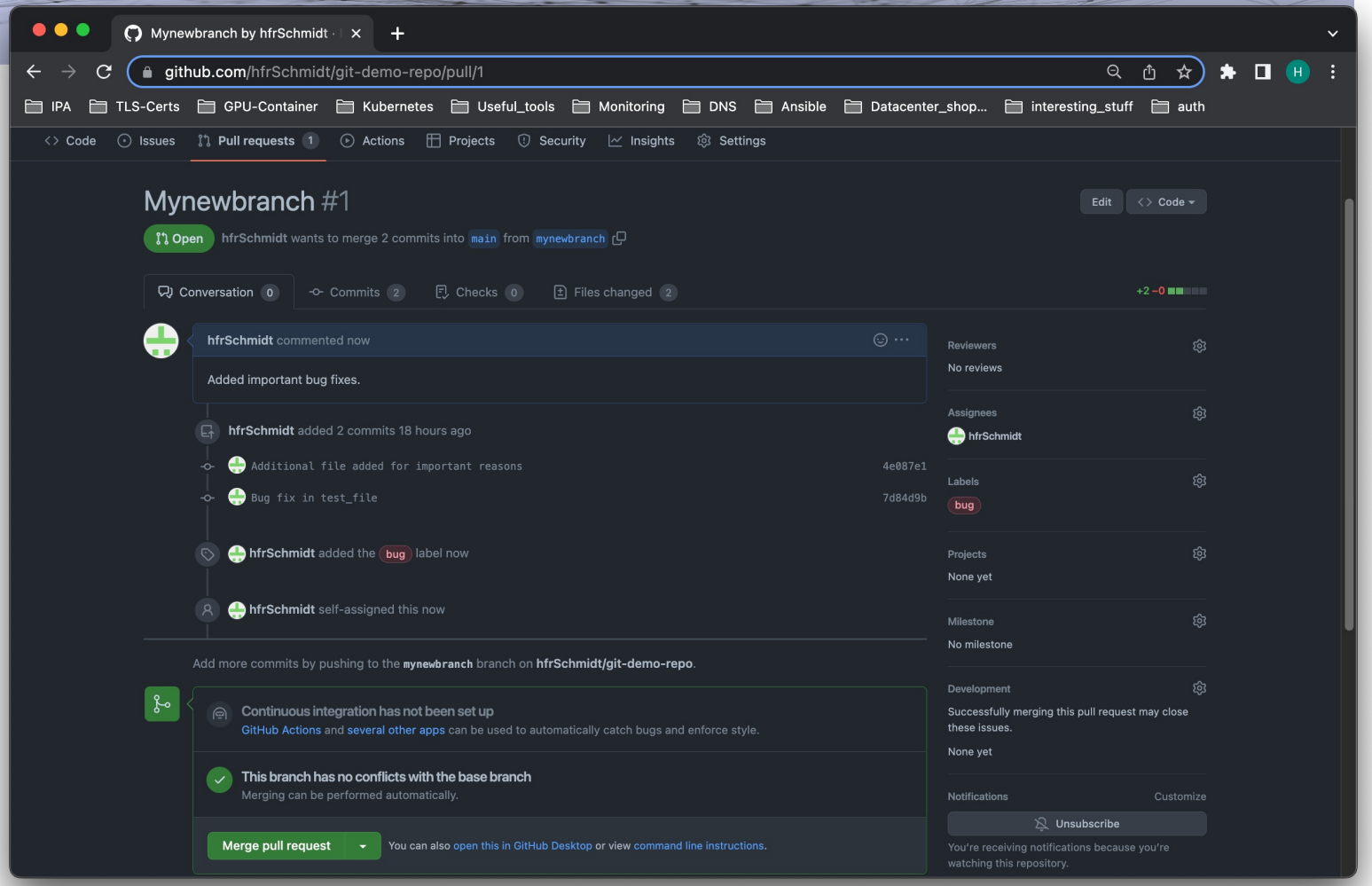
Git in action

- Creating a pull request



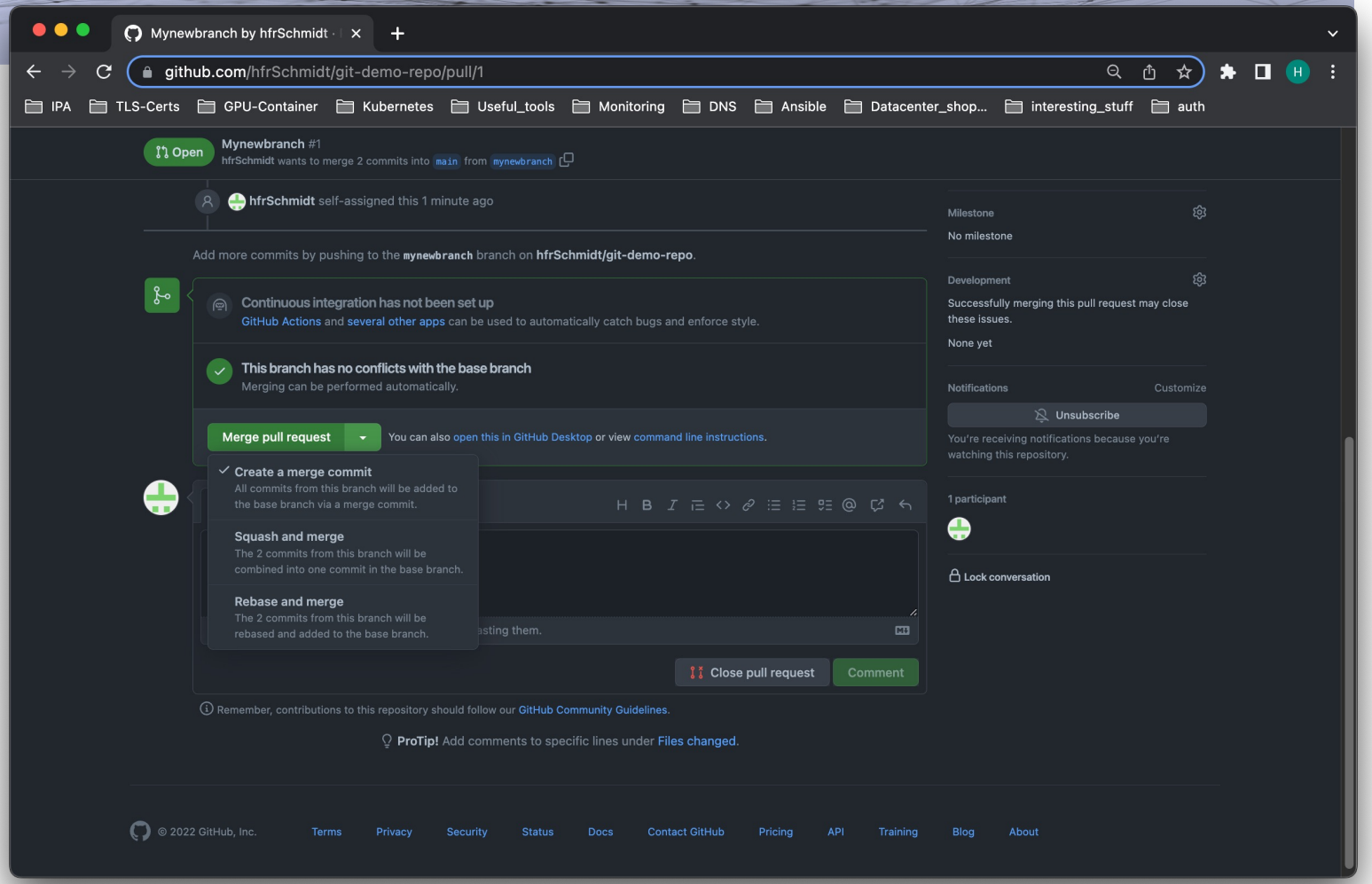
Git in action

- Creating a pull request



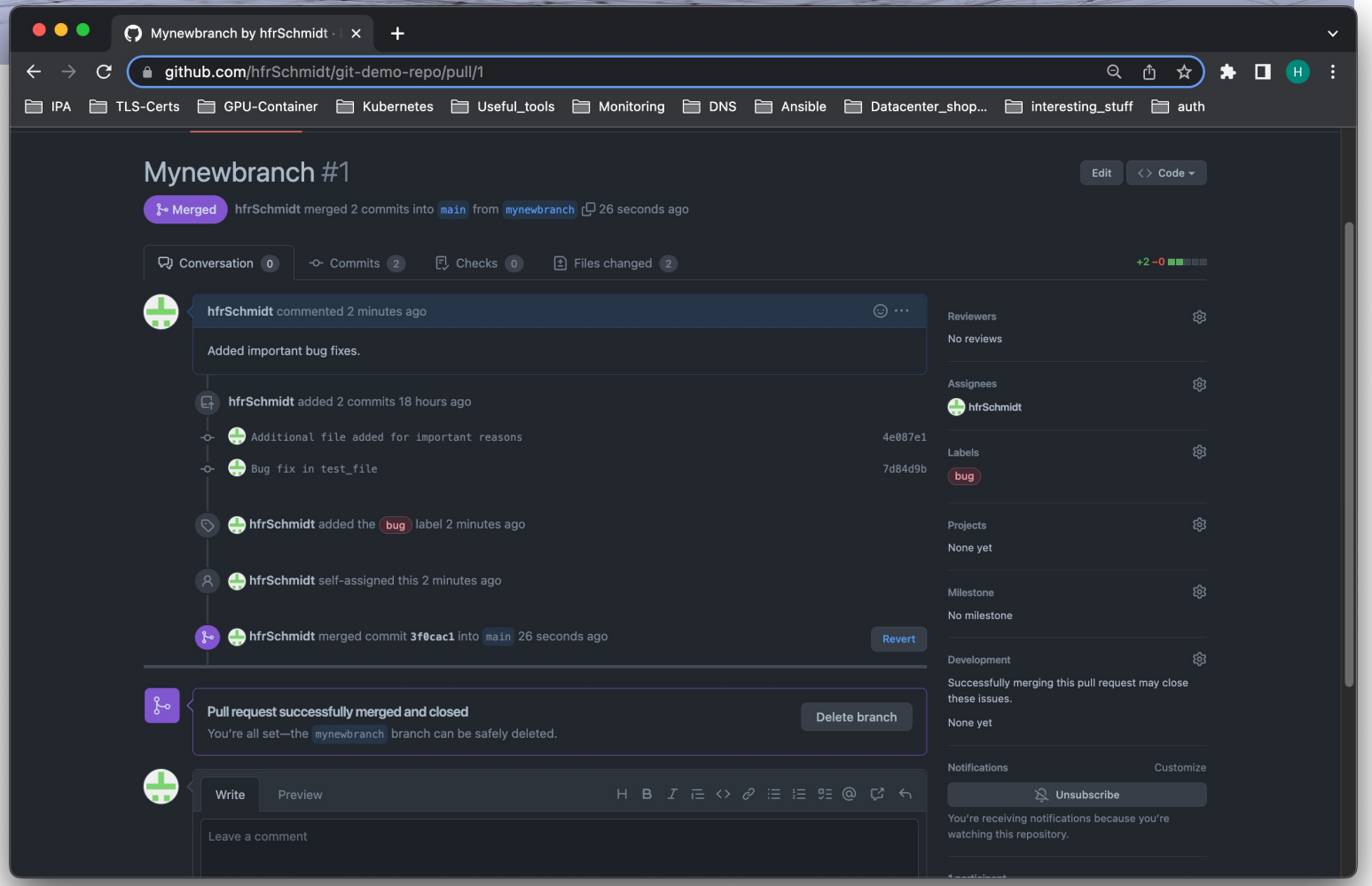
Git in action

- Creating a pull request: PR merge strategy



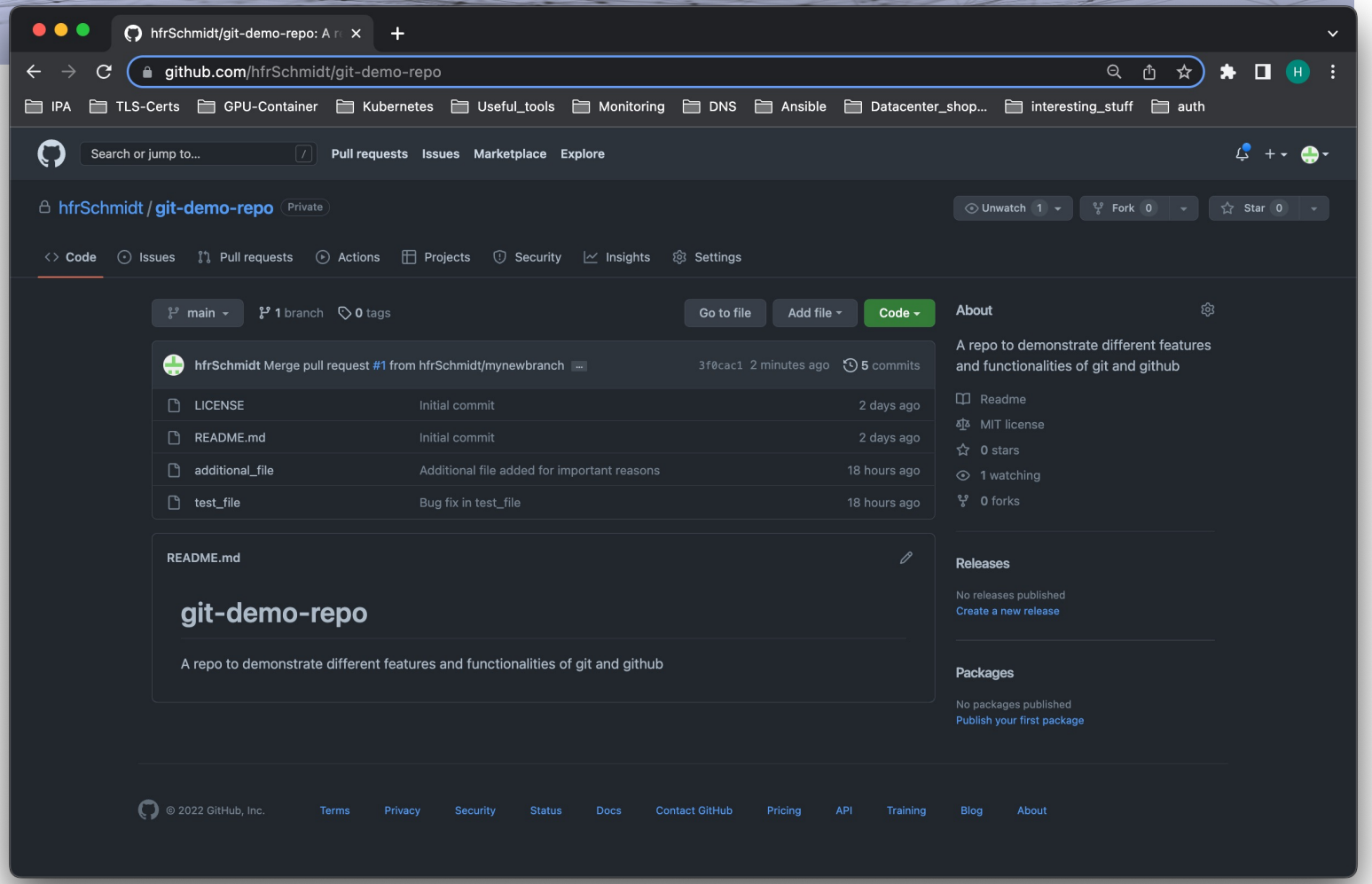
Git in action

- Merged PR



Git in action

- Merged PR



Software development workflows

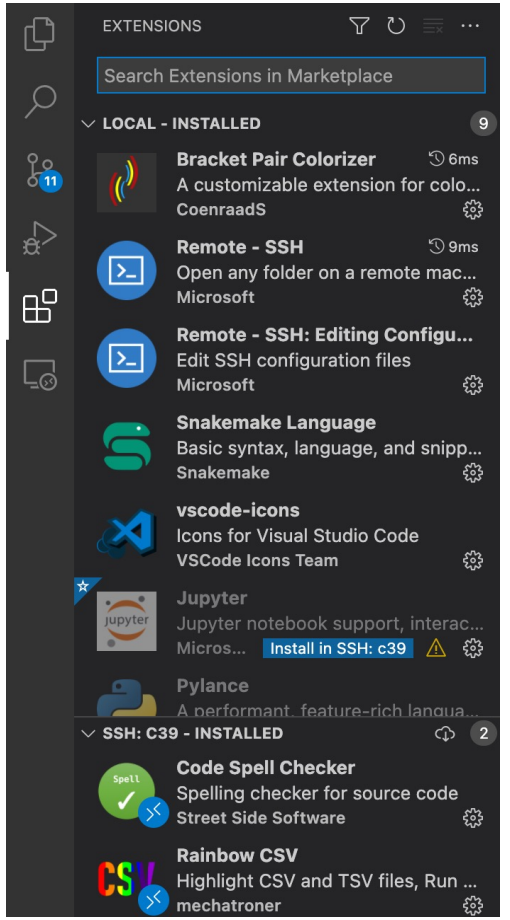
- Visual Studio Code:
  - Remote development extension
  - (Remote development container extension)
- Edit files directly on the server (not entirely recommended):
  - VI
  - VIM
  - Nano
- Edit files locally and sync them with the remote server (not recommended)

Software development workflows

- Example:
  - VS Code Remote SSH extension
  - VS Code extensions:
    - Code Spell Checker
    - Rainbow CSV
    - Bracket pair colorizer
    - Git lens
  - Conda / Mamba environments

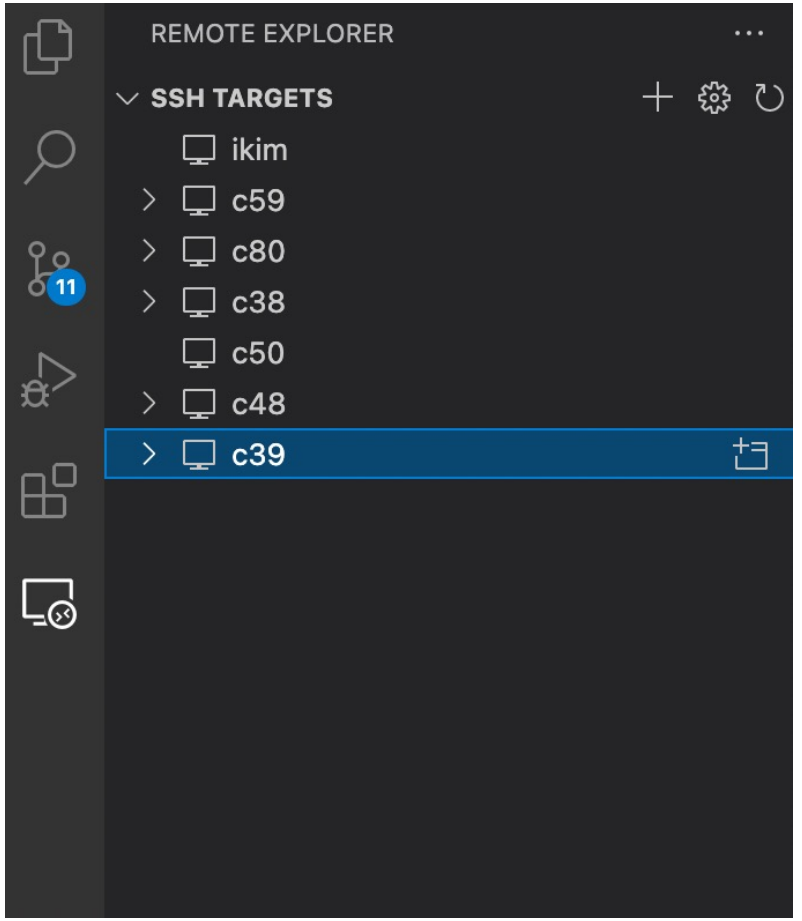
Software development workflows

- Different extensions that improve the usability of VSCode
- Helpful code corrections



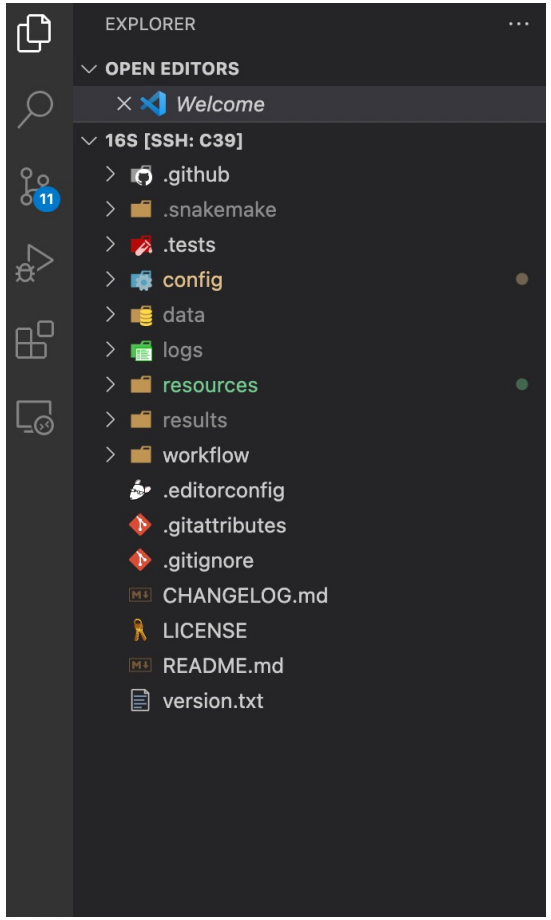
Software development workflows

- Remote-SSH extension
- Showing the server nodes that are saved in the ssh-config
- Makes it easier to connect to the nodes



Software development workflows

- Cloned repository of our 16S pipeline
- Branch symbol shows changes made to the code in comparison to main-branch
- Tracks the changes automatically



Dependency management

- Common problem: „Your code does not run on my system.“
  - Potential causes: Different versions of libraries, missing libraries, different interpreter versions, …
- Usecases for fixing dependencies in one‘s code:
  - Reproducibility (especially in science)
  - Stability
- How can you do this?
  - Don‘t update your OS (very dangerous! Do not do this!)
  - Use a dependency / environment / package manager
    - Pipenv
    - Conda
    - …?

Dependency management

- Conda
  - Package manager + environment manager
- Conda vs Mamba
- Anaconda vs Miniconda vs Mambaforge
- You can install mambaforge via the links at: https://github.com/conda-forge/miniforge#mambaforge

Dependency management

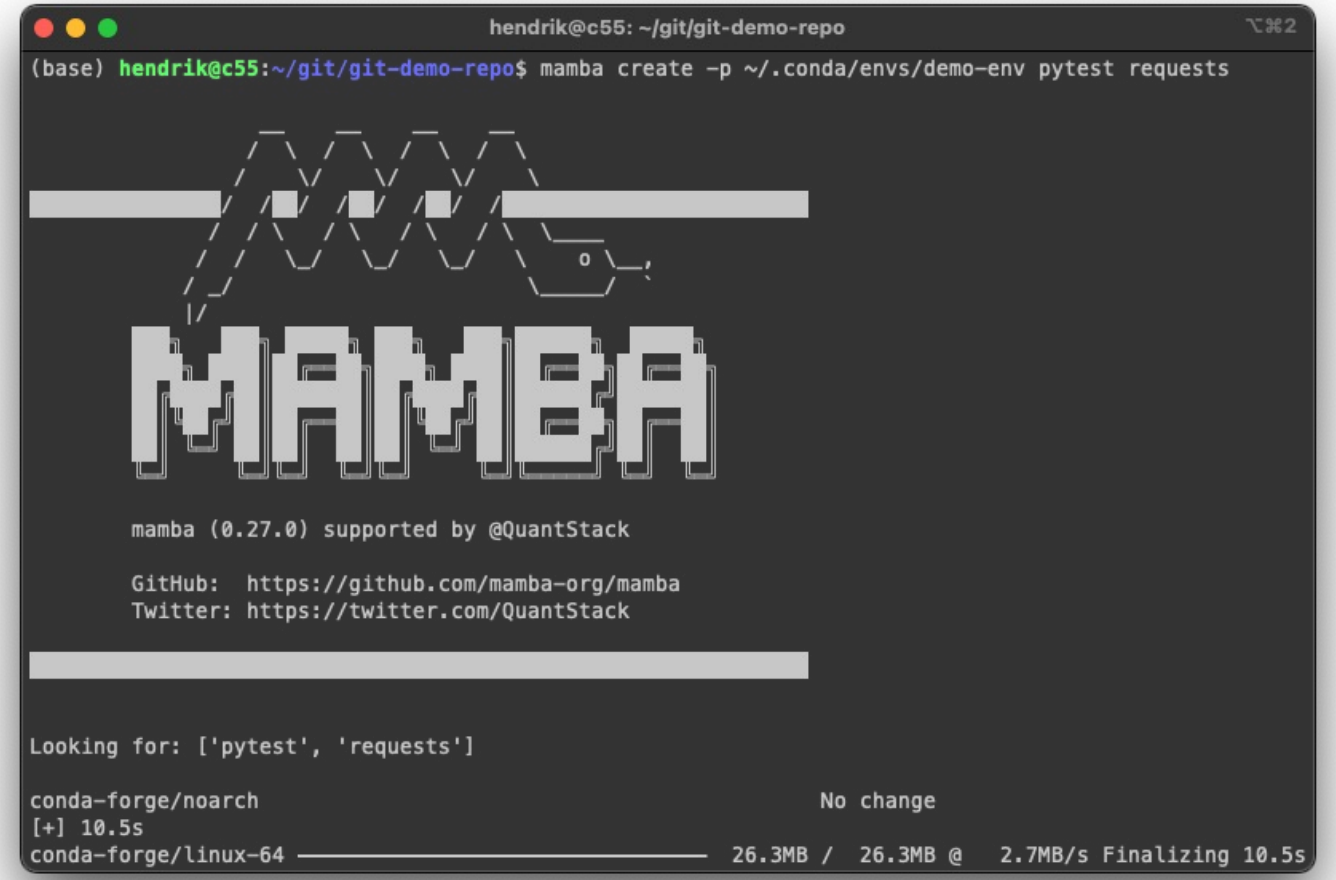
- For example: a basic python app

git > demo > git-demo-repo > test_sample.py > ...

```python
import pytest
import requests

class FunctionalityClass:
    def get_gh_public_timeline(self):
        return requests.get('https://api.github.com/events')

    def post_httpbin_org(self):
        return requests.post('https://httpbin.org/post', data={'key': 'value'})

class TestFuctionalityClass:
    def test_get_gh_public_timeline_success(self):
        response = FunctionalityClass().get_gh_public_timeline()
        assert(response.status_code == requests.codes.ok)

    def test_post_httpbin_org_success(self):
        response = FunctionalityClass().post_httpbin_org()
        assert(response.status_code == requests.codes.ok)

```

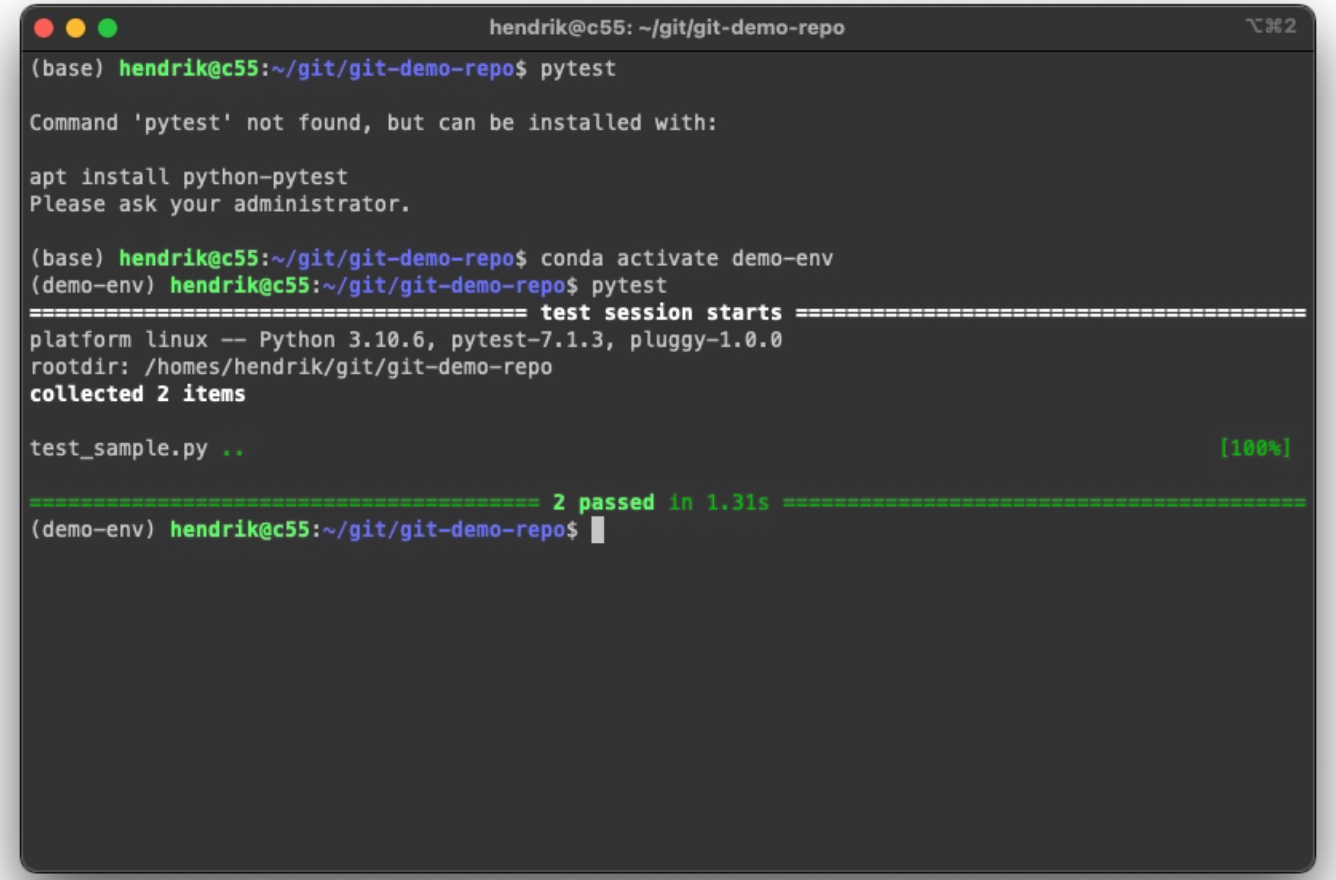
Dependency management

- Create an environment with mamba



Dependency management

- Test the environment



Containers and VMs

- Large differences concerning:
  - Iteration speed
  - Storage overhead
  - Security / isolation
  - Data storage