CSE 403

Software Engineering

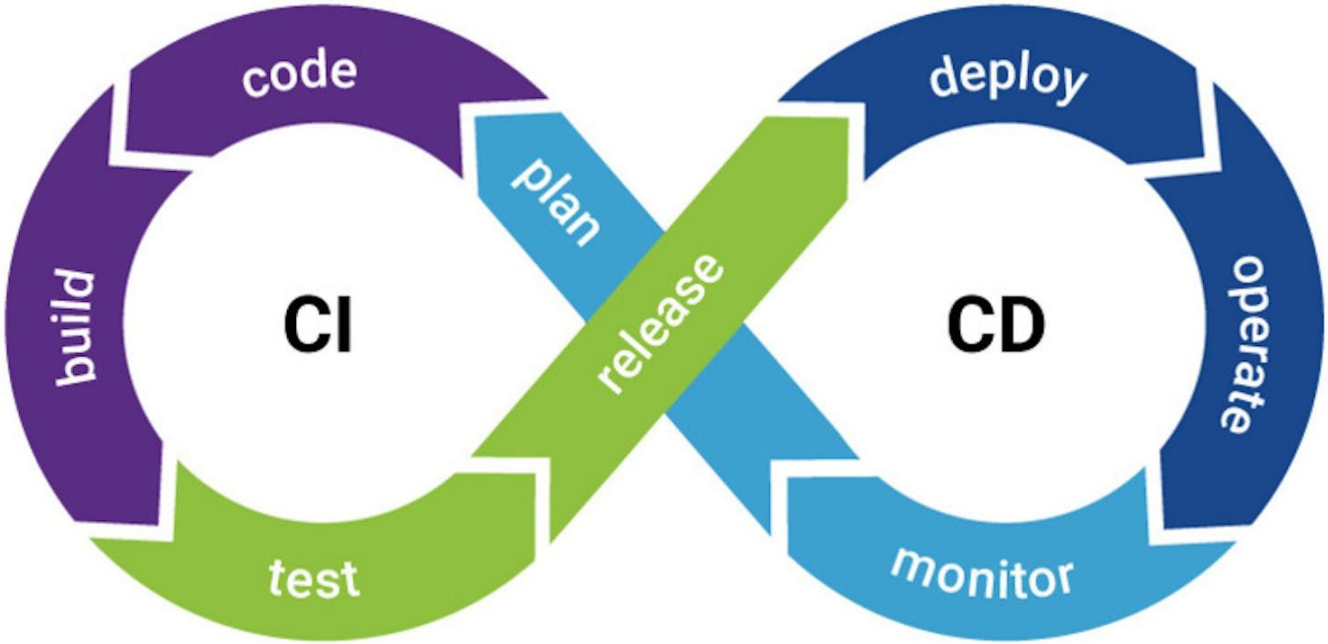
Spring 2023

#15: CI/CD

Logistics

WEEK 5		
04/24	L: Build Systems	
04/25	T:	DUE: DnA!!!
04/26	L: Testing	Testing & CI/CD (TCC)
04/27	P:	
04/28	L: CI/CD	
WEEK 6		
05/01	L: Test Coverage	
05/02	T:	DUE: TCC!!!
05/03	L: Mutation Testing	Alpha Release (R1)
05/04	P:	
05/05	LX: Code Defenders	

Continuous Integration & Continuous Delivery



CI/CD: What's the difference?

Continuous Integration (CI)

- Integrates code into a shared repository.
- Builds/tests each change automatically.
- Complements local developer workflows (subset of tests vs. all tests).

Continuous Deployment (CD)

- Builds on top of CI.
- Software can be deployed at any time.
- Automatically pushes changes to production.

CI/CD: What's the difference?

Continuous Integration (CI)

- Integrates code into a shared repository.
- Builds/tests each change automatically.
- Complements local developer workflows (subset of tests vs. all tests).

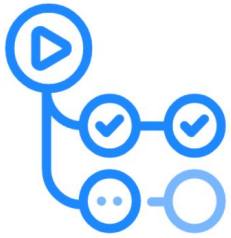
Continuous Deployment (CD)

- Builds on top of CI.
- Software can be deployed at any time.
- Automatically pushes changes to production.

403 focuses on establishing good CI practices.

Continuous Integration & Delivery Tools

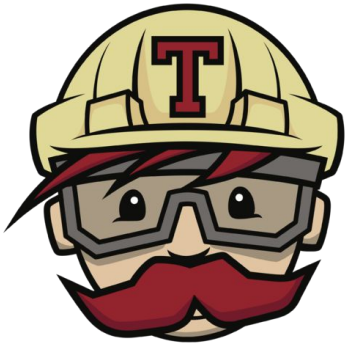
Jenkins



GitHub Actions

AWS CodePipeline

Bitbucket Pipelines



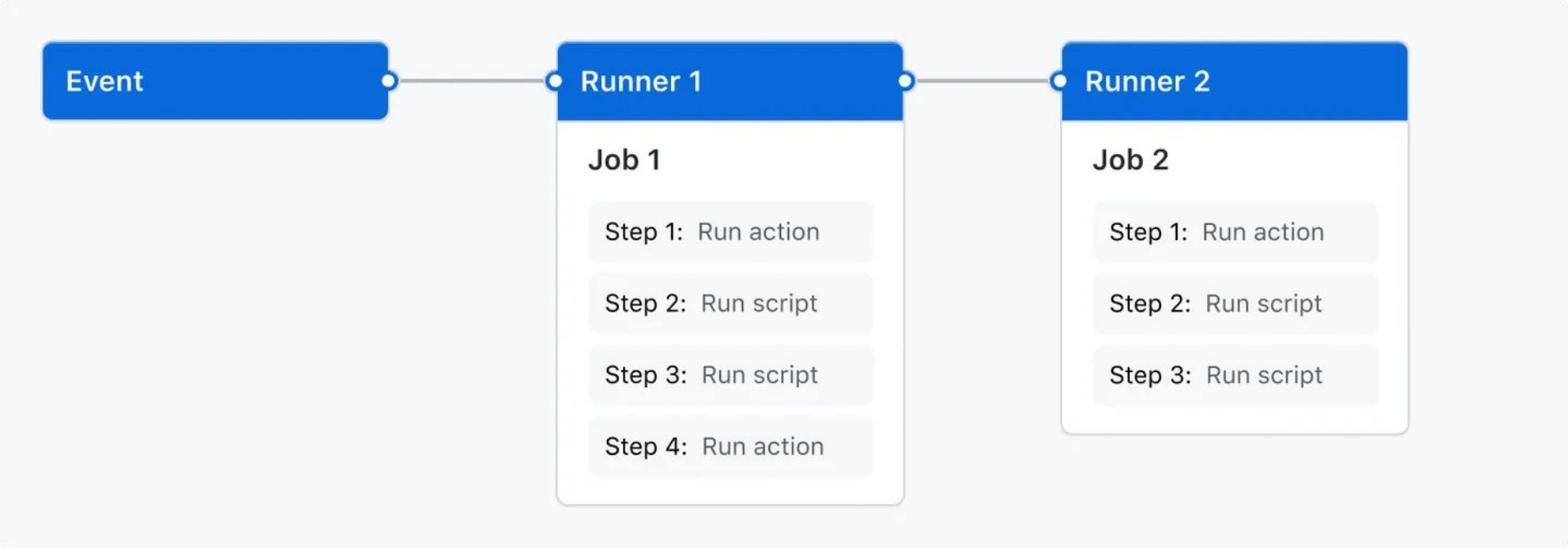
Travis CI

GitLab

Azure Pipelines

circleci

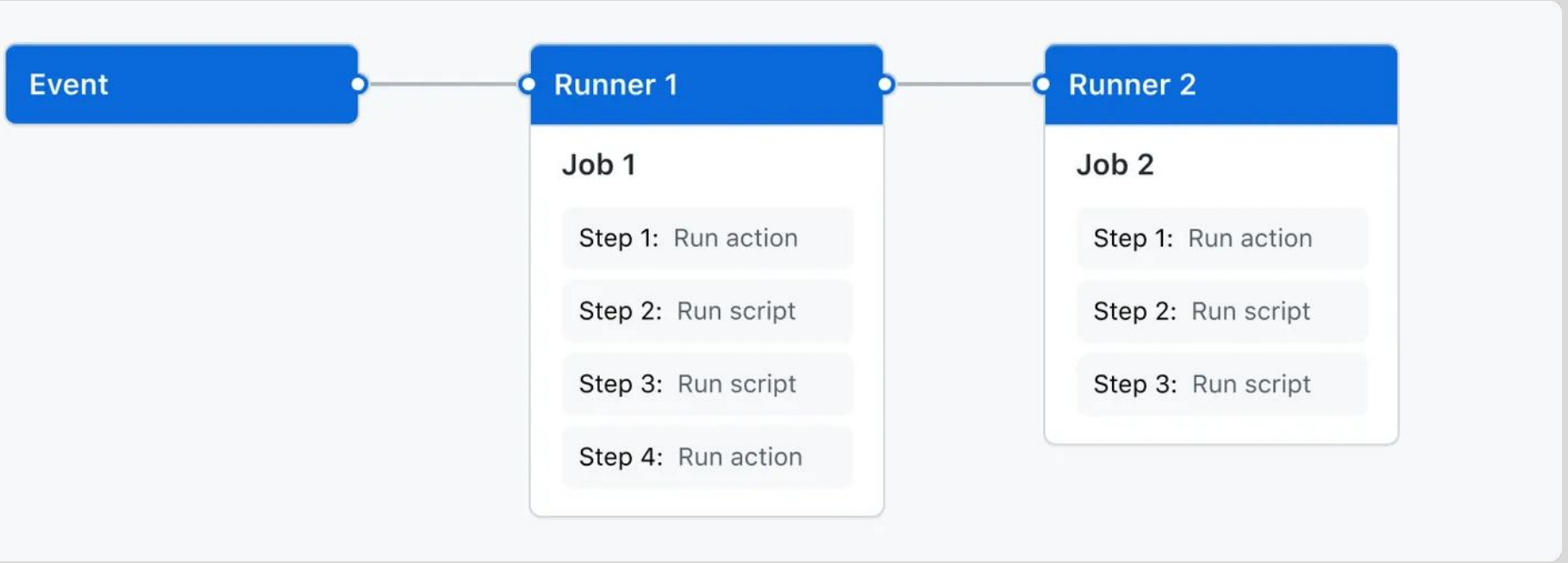
CI (w/ GitHub Actions) Basics



Terminology: Workflow (pipeline), Event, Job, Action, Runner

https://docs.github.com/en/actions/learn-github-actions/understanding-github-actions

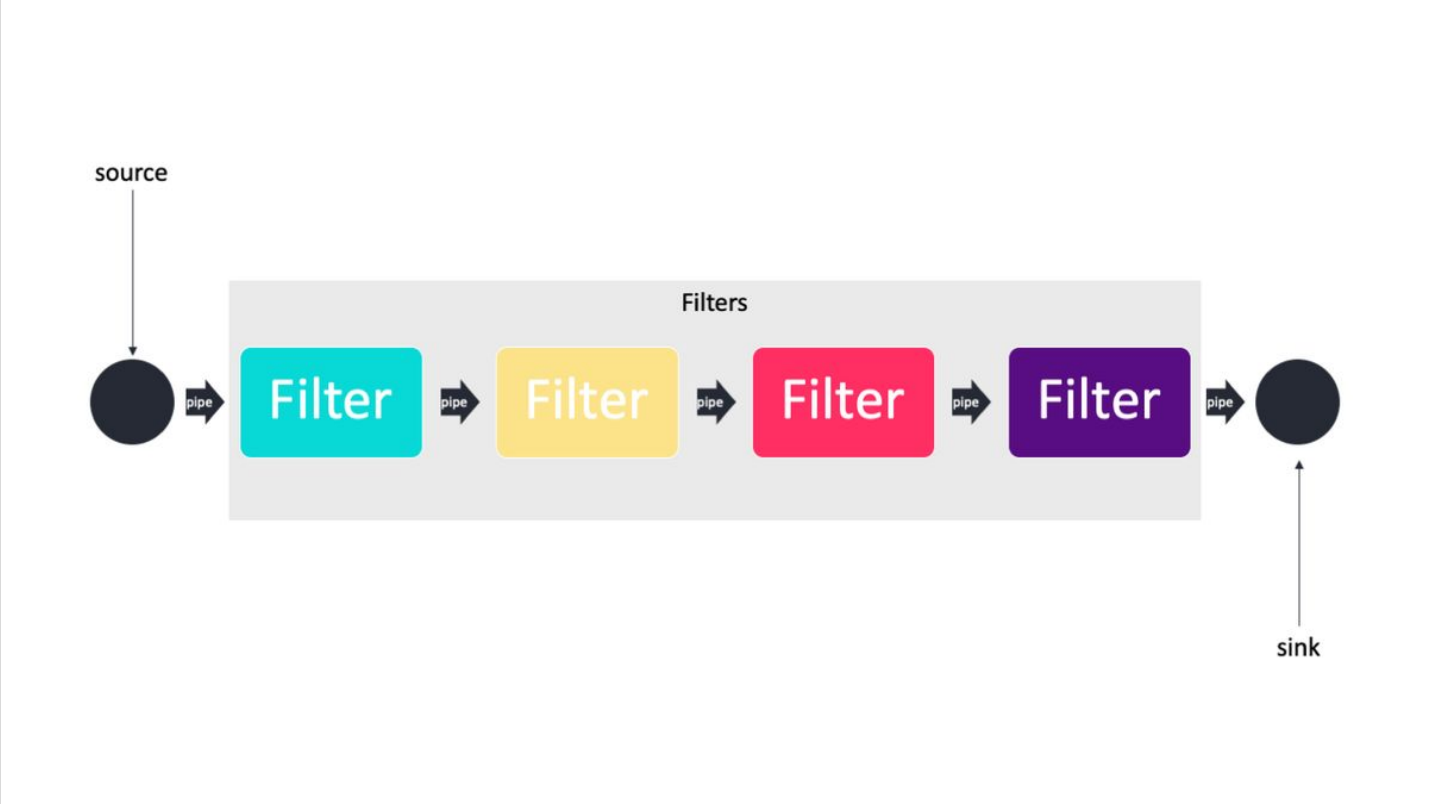
CI (w/ GitHub Actions) Basics



Terminology: Workflow (pipeline), Event, Job, Action, Runner

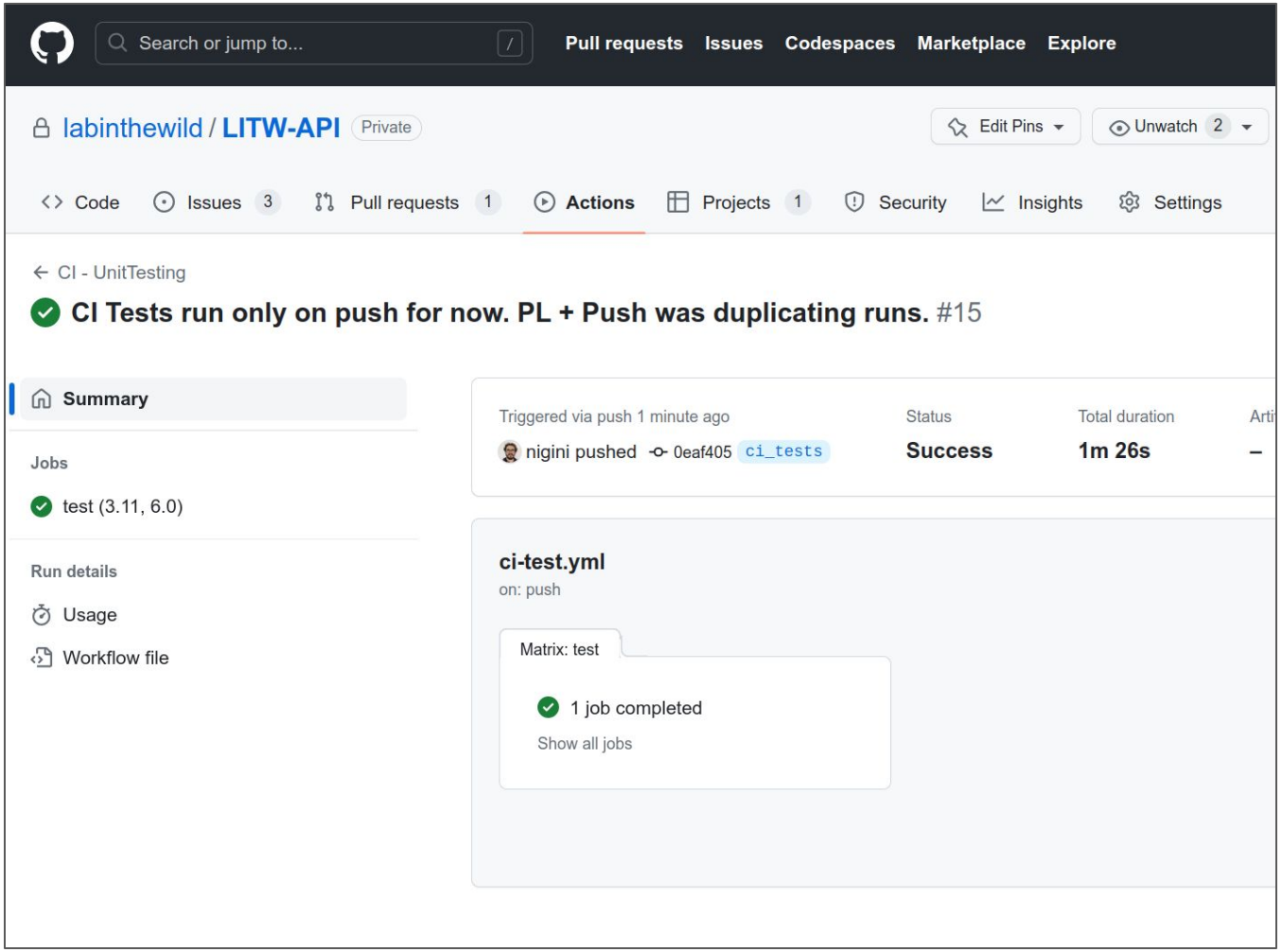
BTW: What SW Architecture do you think CI/CD uses?

SW Architecture #1: Pipe and Filter



The pipe-and-filter **architecture** doesn't specify the **design** or **implementation** details of the individual components (the filters)!

Live example: CI in action



Live example: CI in action

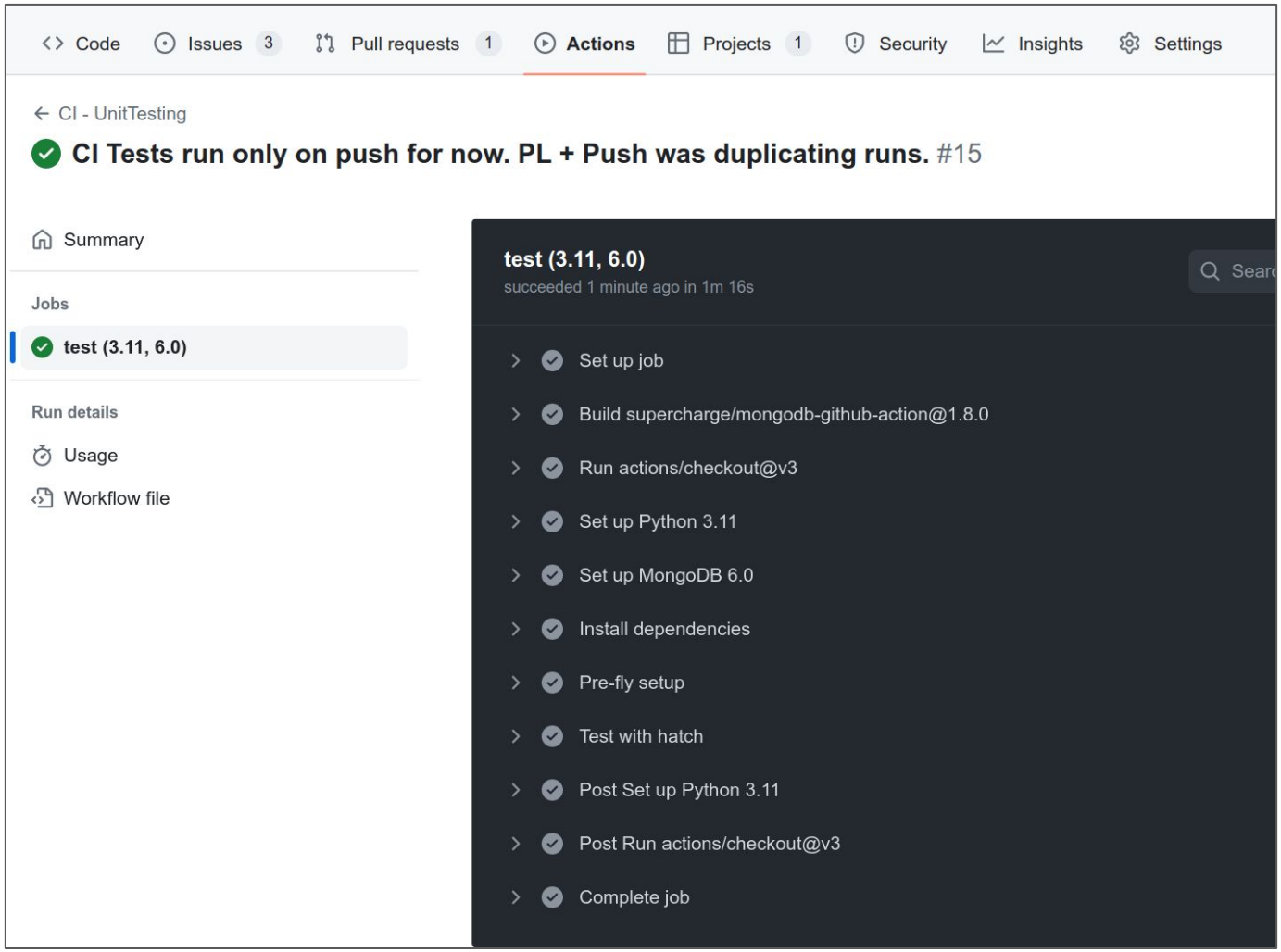
```yaml
name: CI - UnitTesting
on: [push]
jobs:
  test:
    runs-on: ubuntu-latest
    strategy: <2 keys>

    steps:
    - uses: actions/checkout@v3
    - name: Set up Python ${{ matrix.python-version }}
      uses: actions/setup-python@v3
      with: <1 key>
    - name: Set up MongoDB ${{ matrix.mongodb-version }}
      uses: supercharge/mongodb-github-action@1.8.0
      with: <1 key>
    - name: Install dependencies
      run: python3 -…tall hatch
    - name: Pre-fly setup
      run: cp $GITHU…GITHUB_ENV
    - name: Test with hatch
      run: |
        hatch run test:test
```

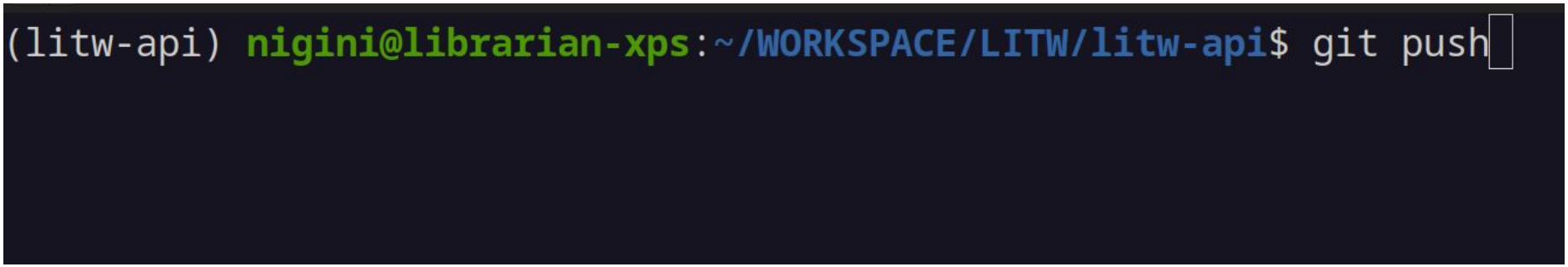
Reuse other people's work!

Remember L13?
One command!

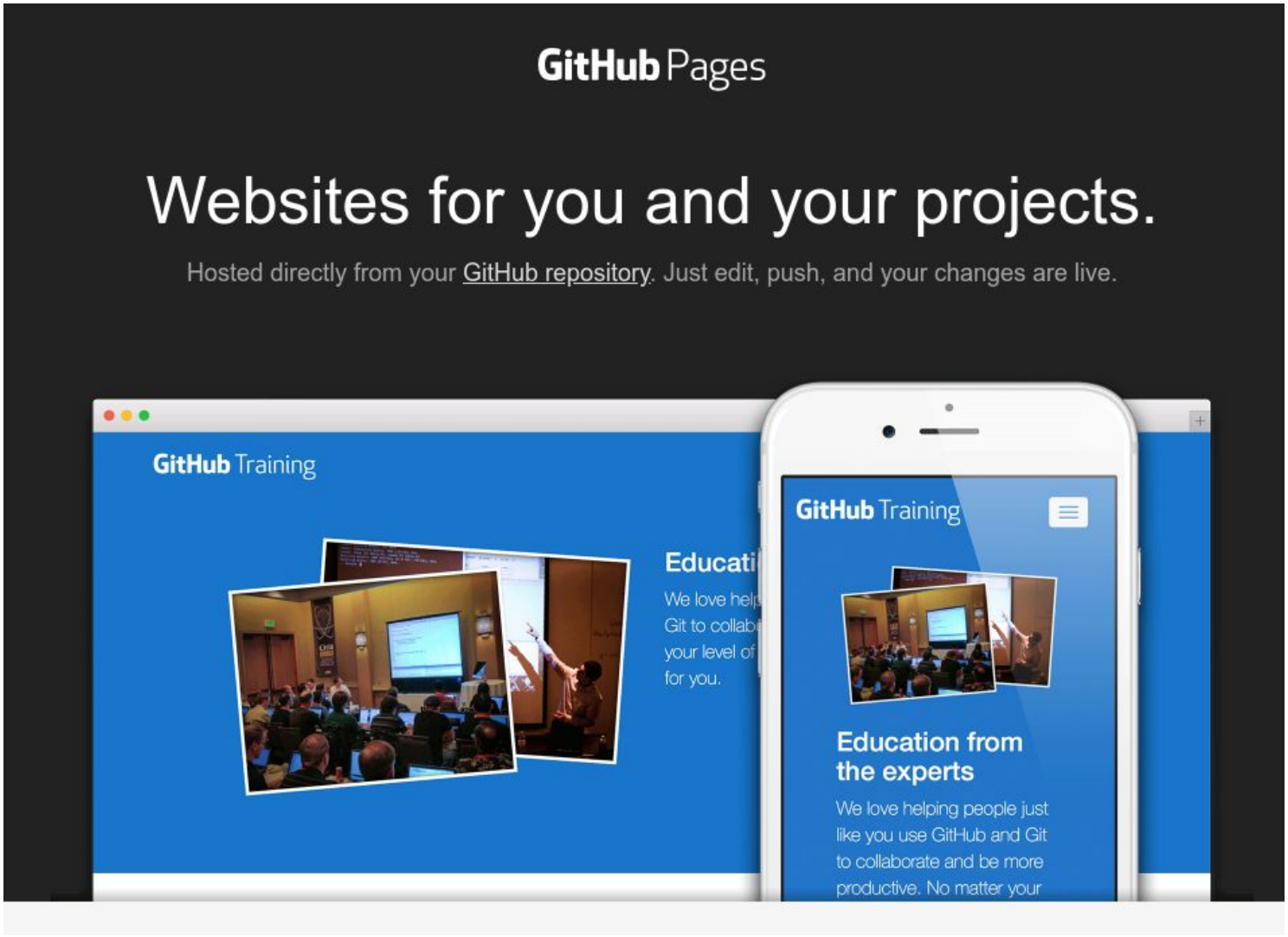
Live example: CI in action



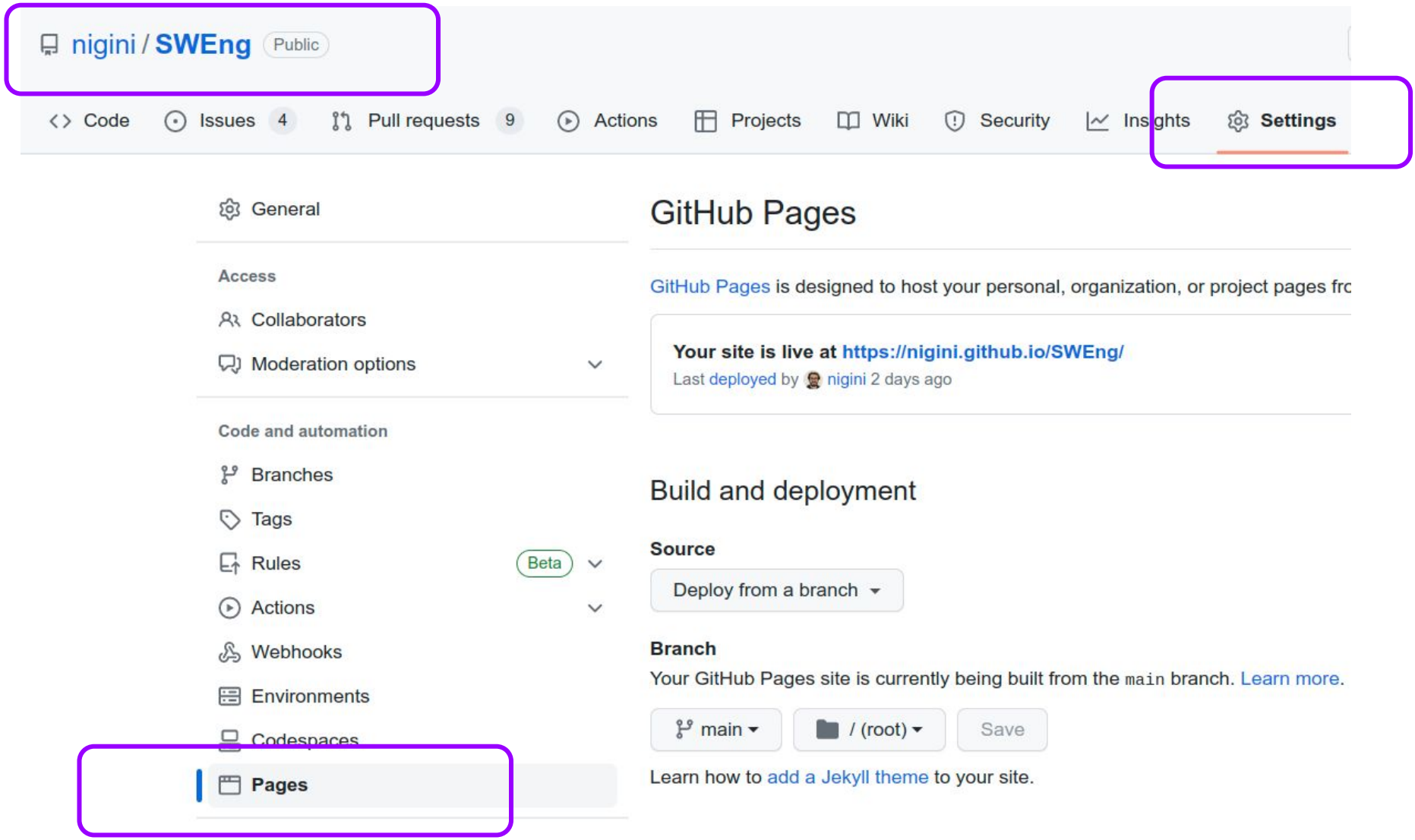
Live example: CI in action



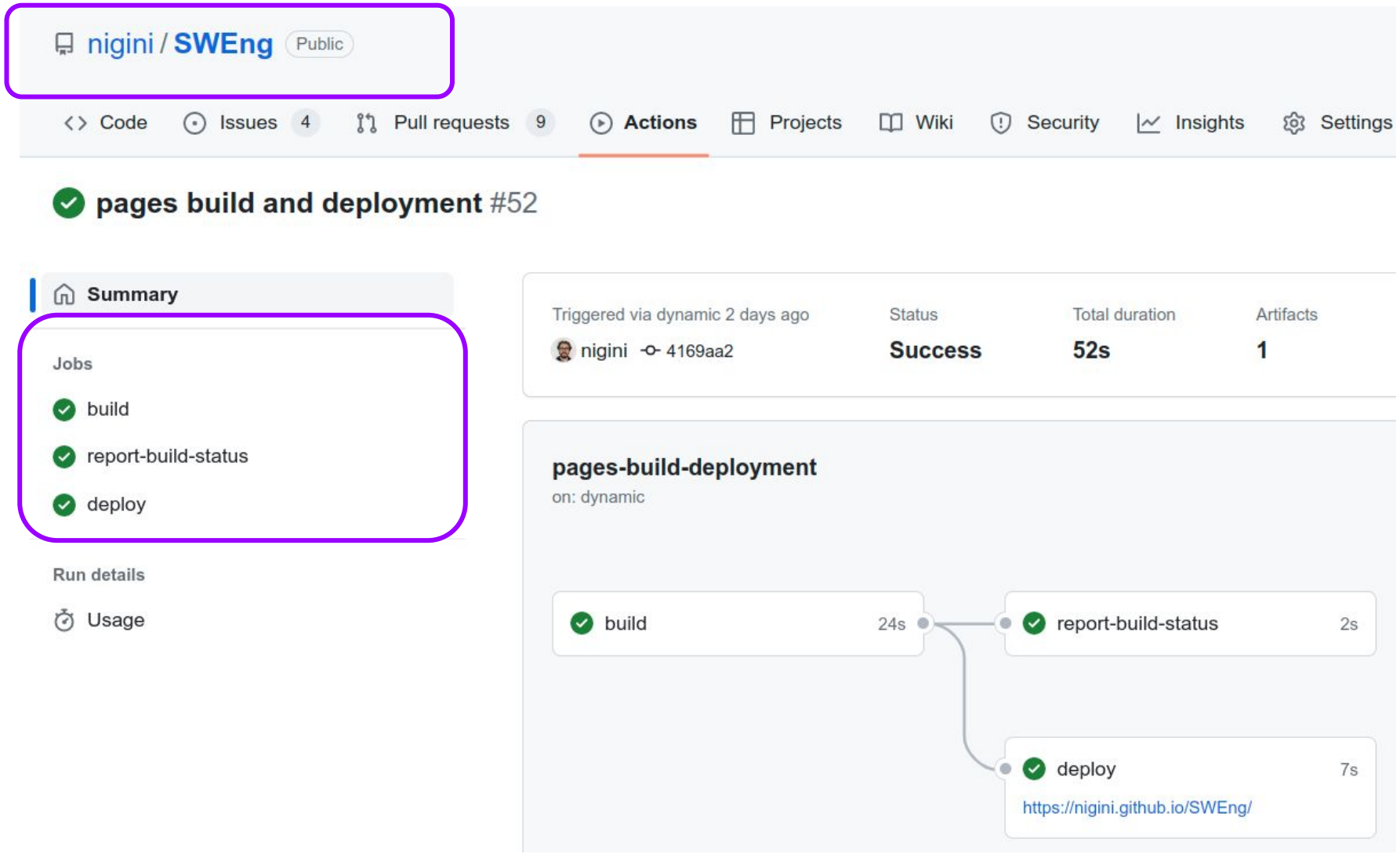
Example: CD in action



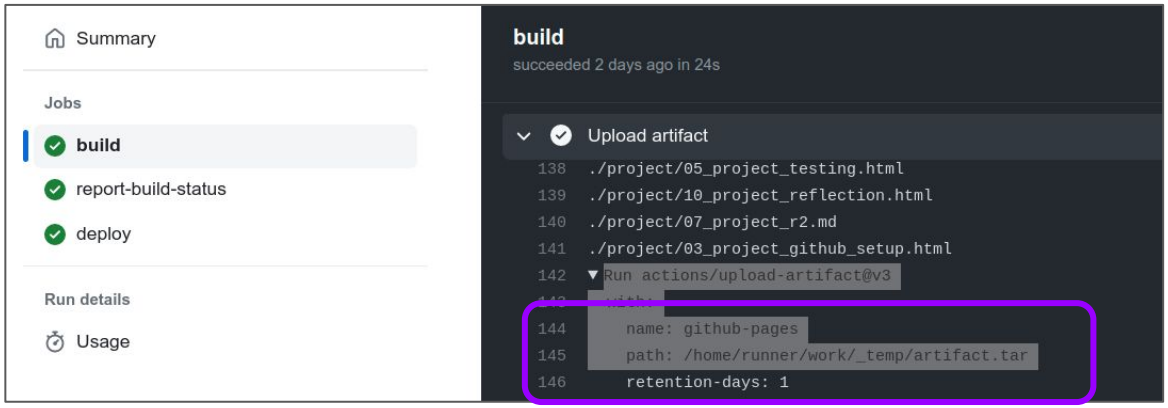
Example: CD in action

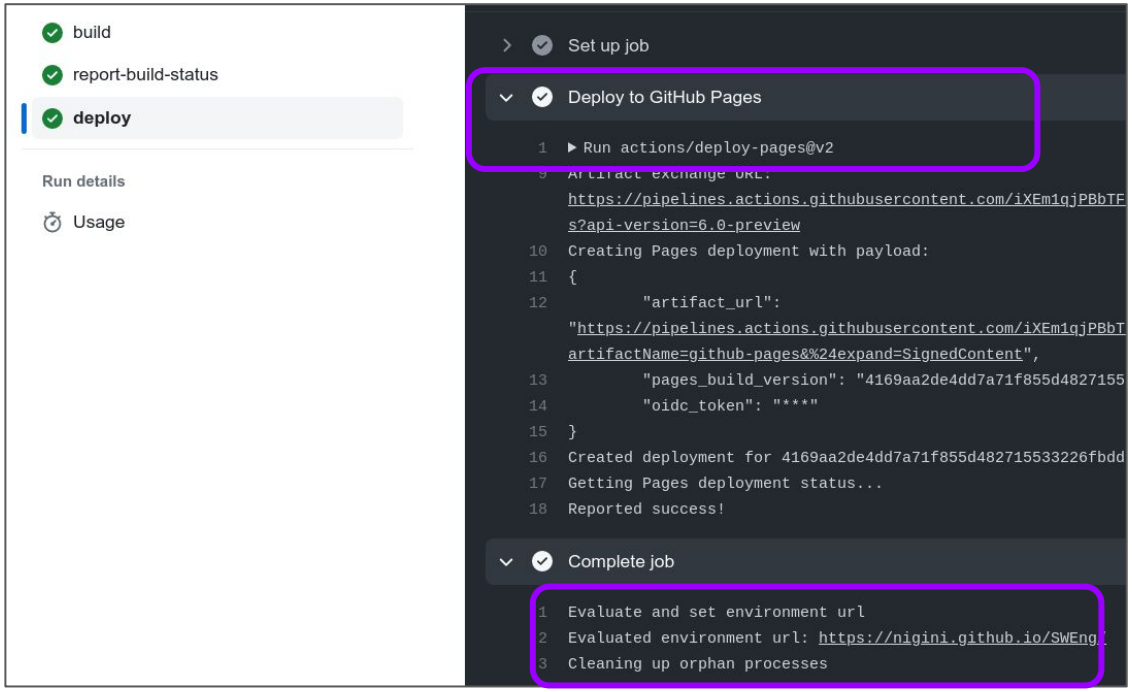


Example: CD in action

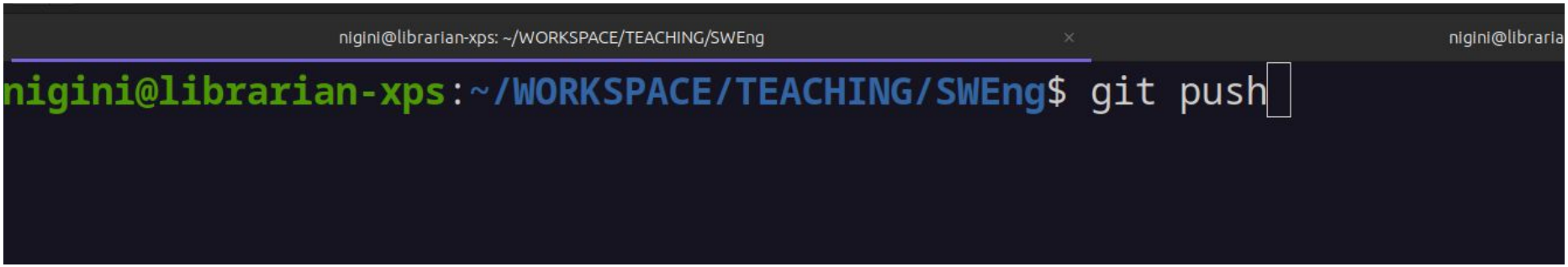


Example: CD in action

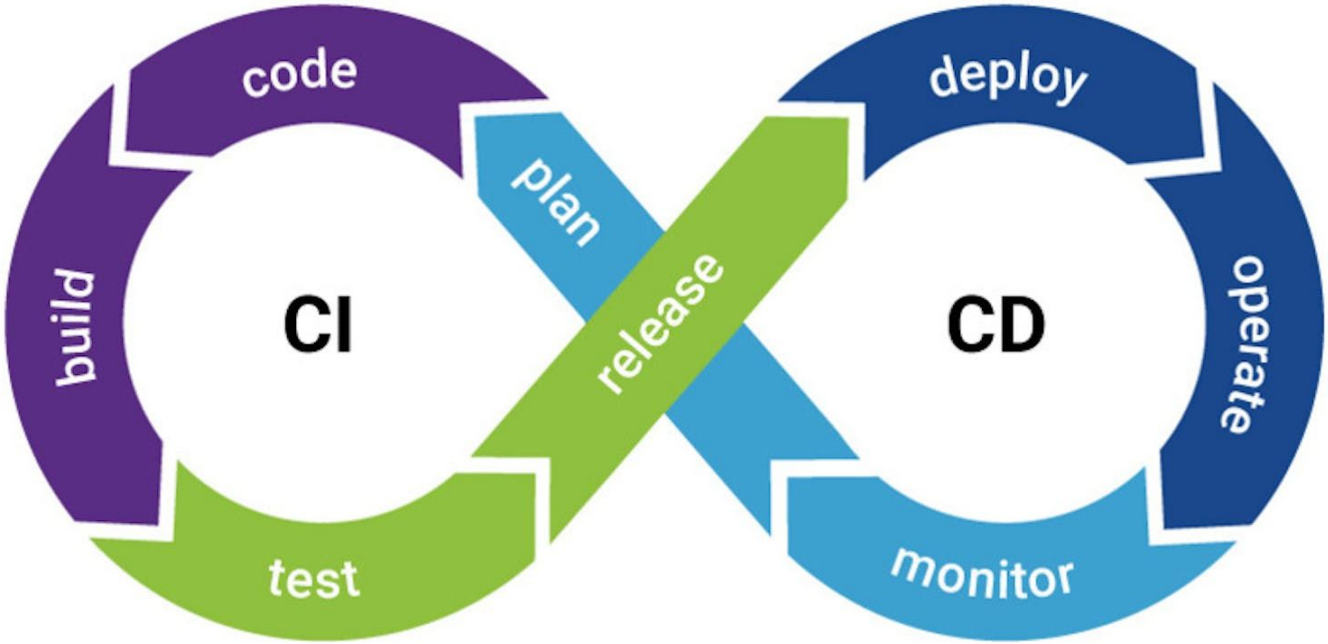




Example: CD in action



Continuous Integration & Continuous Delivery



Questions, please!!!