

# Setting up a CI/CD Pipeline in GitHub Actions

Introduction to Python Software  
Development on GitHub 2023

Lyda Hill Department of Bioinformatics  
UT Southwestern Medical Center



# GitHub Actions Dashboard

The screenshot shows the GitHub Actions Dashboard for the repository 'TheDeanLab / ASLM'. The top navigation bar includes links to Code, Issues (36), Pull requests (2), Discussions, Actions (selected), Projects (1), Security (49), Insights, and Settings. A search bar is located on the right. The left sidebar lists various workflow categories: All workflows, Build Docs, Build Wheel, Lint, pages-build-deployment, Tests (selected), Management, Caches, Deployments, and Runners (Beta). The main content area is titled 'Tests' and shows a list of workflow runs for the 'push\_checks.yaml' file. It indicates there are 1,232 workflow runs. A message states: 'This workflow has a workflow\_dispatch event trigger.' with a 'Run workflow' button. The list of runs includes:

Event	Status	Branch	Actor
Make the XML a bit nicer	Success	zacsimile:xml-advances	18 hours ago
Expose BDV data source as array	Success	zacsimile:ds-load	yesterday
Expose BDV data source as array	Success	zacsimile:ds-load	yesterday

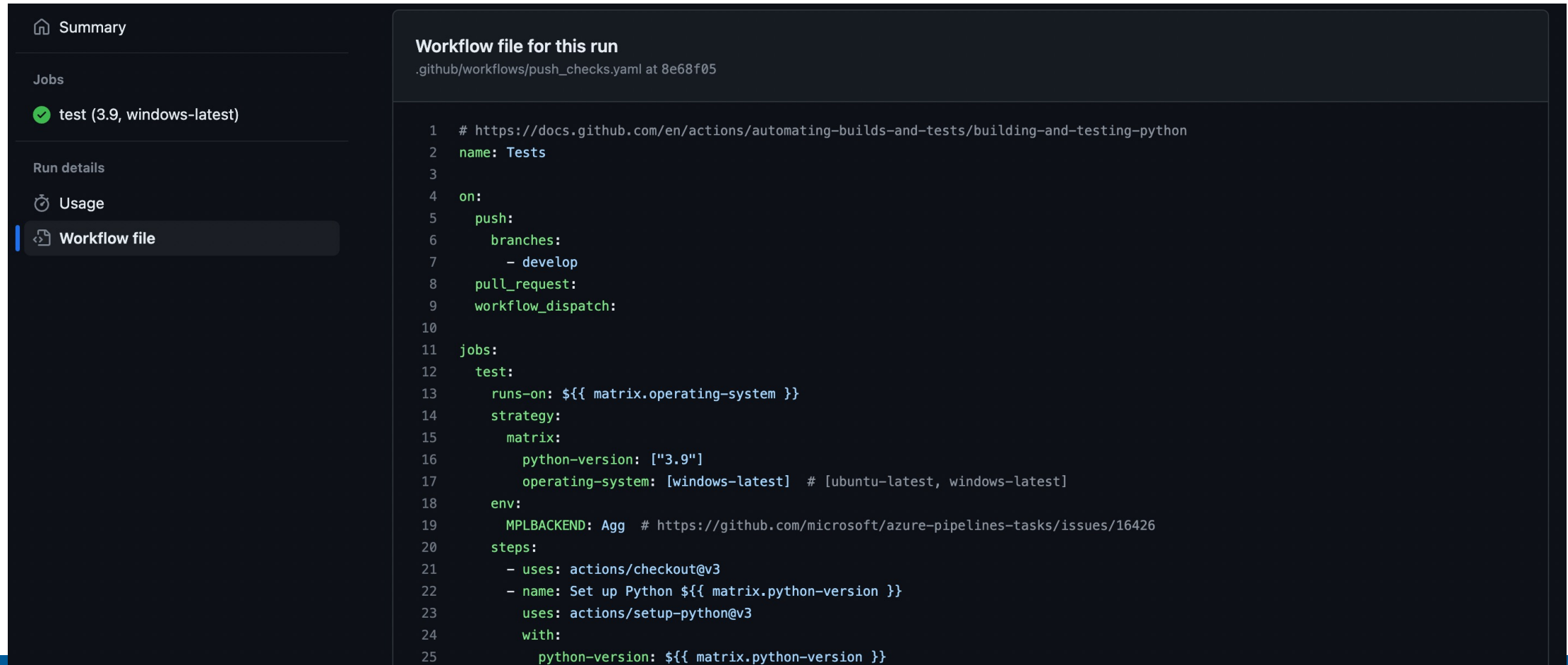
Each run entry also includes a description of the test (e.g., 'Tests #1792: Pull request #643 opened by zacsimile') and a duration (e.g., '31m 46s').

# GitHub Actions Workflow Example

The screenshot shows a GitHub Actions workflow run for the repository "Make the XML a bit nicer #1792". The workflow is named "test (3.9, windows-latest)" and has successfully completed 17 hours ago in 31m 31s. The left sidebar contains navigation links: Summary, Jobs, Run details, Usage, and Workflow file. The "Jobs" section is expanded, showing the job "test (3.9, windows-latest)". The main area displays a list of steps in the job, each with a status icon, a description, and a duration.

Step	Status	Duration
Set up job	✓	2s
Run actions/checkout@v3	✓	8s
Set up Python 3.9	✓	0s
Install dependencies	✓	3m 38s
Test with pytest	✓	27m 36s
Codecov	✓	3s
Post Set up Python 3.9	✓	0s
Post Run actions/checkout@v3	✓	1s
Complete job	✓	0s

# GitHub Actions Workflow Example



The screenshot shows the GitHub Actions interface for a workflow. On the left sidebar, the 'Workflow file' tab is selected. The main area displays the workflow file content, which is a YAML configuration for a workflow named 'Tests'. The workflow is triggered on push to the 'develop' branch, pull requests, or workflow dispatch. It consists of a single job named 'test' that runs on the latest version of Windows. The job uses a matrix strategy to test with Python 3.9. The steps include checking out the code, setting up Python, and installing dependencies.

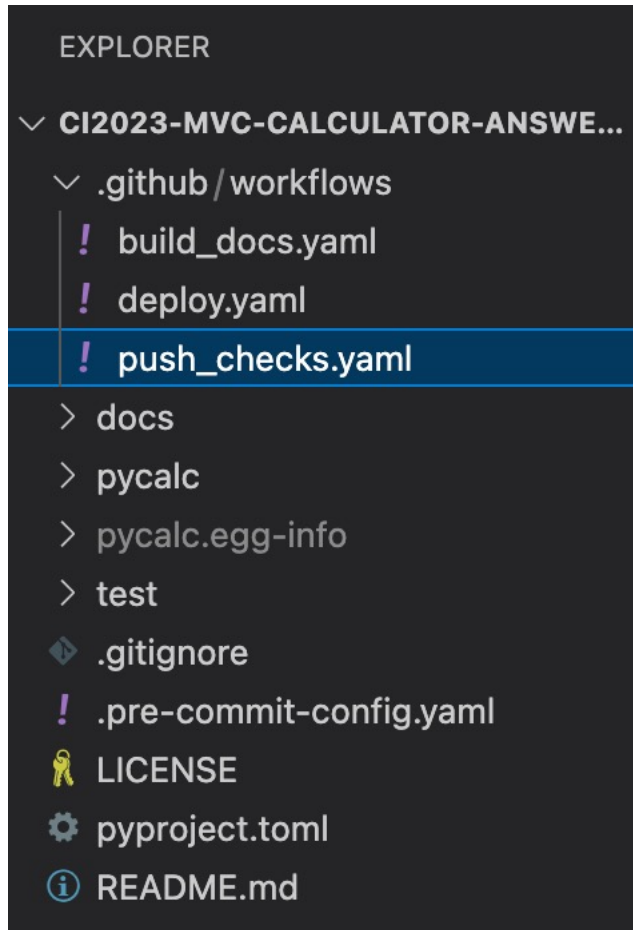
**Workflow file for this run**  
github/workflows/push\_checks.yaml at 8e68f05

```
1 # https://docs.github.com/en/actions/automating-builds-and-tests/building-and-testing-python
2 name: Tests
3
4 on:
5   push:
6     branches:
7       - develop
8   pull_request:
9   workflow_dispatch:
10
11 jobs:
12   test:
13     runs-on: ${ matrix.operating-system }
14     strategy:
15       matrix:
16         python-version: ["3.9"]
17         operating-system: [windows-latest] # [ubuntu-latest, windows-latest]
18     env:
19       MPLBACKEND: Agg # https://github.com/microsoft/azure-pipelines-tasks/issues/16426
20     steps:
21       - uses: actions/checkout@v3
22       - name: Set up Python ${ matrix.python-version }
23         uses: actions/setup-python@v3
24         with:
25           python-version: ${ matrix.python-version }
```

# Let's create a GitHub workflow

- You can follow along at <https://docs.github.com/en/actions/quickstart>

# Where does the workflow file go?



- They are stored in your Git repository, under `.github/workflows`
- Create the `.github` folder
- Create the `.github/workflows` folder
- Create `github-actions-demo.yml`

# github-actions-demo.yml

- <https://docs.github.com/en/actions/quickstart>

```
YAML

name: GitHub Actions Demo
run-name: ${ github.actor } is testing out GitHub Actions 🚀
on: [push]
jobs:
  Explore-GitHub-Actions:
    runs-on: ubuntu-latest
    steps:
      - run: echo "🎉 The job was automatically triggered by a ${ github.event_name } event."
      - run: echo "👀 This job is now running on a ${ runner.os } server hosted by GitHub!"
      - run: echo "🔎 The name of your branch is ${ github.ref } and your repository is ${ github.repository }."
      - name: Check out repository code
        uses: actions/checkout@v4
      - run: echo "💡 The ${ github.repository } repository has been cloned to the runner."
      - run: echo "🖨 The workflow is now ready to test your code on the runner."
      - name: List files in the repository
        run: |
          ls ${ github.workspace }
      - run: echo "🍏 This job's status is ${ job.status }."
```

# What are these two lines doing??

```
name: GitHub Actions Demo
run-name: ${{ github.actor }} is testing out GitHub Actions 🚀
```

The screenshot shows the GitHub Actions page for the repository 'TheDeanLab / ASLM'. The 'Actions' tab is selected in the top navigation bar. On the left, the 'Actions' sidebar lists 'All workflows' and several specific workflows: 'Build Docs', 'Build Wheel', 'Lint', 'pages-build-deployment', and 'Tests' (which is currently selected). The main content area is titled 'Tests' and shows the workflow file 'push\_checks.yaml'. It indicates there are '1,232 workflow runs'. A message states: 'This workflow has a workflow\_dispatch event trigger.' with a 'Run workflow' button. Below this, a recent workflow run is displayed with a green checkmark, titled 'Make the XML a bit nicer', triggered by 'zacsimile:xml-advances' 18 hours ago, with a duration of 31m 46s.



# What are these two lines doing??

```
name: GitHub Actions Demo
run-name: ${{ github.actor }} is testing out GitHub Actions 🚀
```

## Actions

New workflow

### All workflows

.github/workflows/greet-everyone.yml

CI

CodeQL

GitHub Actions Demo

## All workflows

Showing runs from all workflows

Filter workflow runs

88 workflow runs

Event ▾

Status ▾

Branch ▾

Actor ▾

✓ octocat is testing out GitHub Actions 🚀

GitHub Actions Demo #1: Commit 9861173 pushed by octocat

octocat-patch-1

2 minutes ago

17s

# What is `${{ }}`?

- It's an expression (see <https://docs.github.com/en/actions/learn-github-actions/expressions>)
- Expressions evaluate what's inside them
- `${{ github.actor }}` evaluates to `octocat` in the example
- `${{ github.actor == "octocat" }}` evaluates to `true`

# What does `on: [push]` do?

- It triggers the jobs under `jobs` when code is `git` pushed to the repository
- There are other events. For example, this will trigger a job
  - on a `git push` to the `main` branch
  - when a Pull Request is opened
  - if we hit "Run workflow" in the Actions dashboard

```
on:  
  push:  
    branches:  
      - main  
  pull_request:  
  workflow_dispatch:
```

# Defining the jobs

```
jobs:
  Explore-GitHub-Actions:
    runs-on: ubuntu-latest
    steps:
      - run: echo "🚀 The job was automatically triggered by a ${github.event_name} event."
```

- What is jobs?

# Defining the jobs

```
jobs:
  Explore-GitHub-Actions:
    runs-on: ubuntu-latest
    steps:
      - run: echo "🚀 The job was automatically triggered by a ${github.event_name} event."
```

- What is `jobs`? The key that holds all the jobs that run in the workflow file.

# Defining the jobs

```
jobs:
  Explore-GitHub-Actions:
    runs-on: ubuntu-latest
    steps:
      - run: echo "🚀 The job was automatically triggered by a ${github.event_name} event."
```

- What is `jobs`? The key that holds all the jobs that run in the workflow file.
- What is `Explore-GitHub-Actions`?

# Defining the jobs

```
jobs:
  Explore-GitHub-Actions:
    runs-on: ubuntu-latest
    steps:
      - run: echo "🚀 The job was automatically triggered by a ${github.event_name} event."
```

- What is `jobs`? The key that holds all the jobs that run in the workflow file.
- What is `Explore-GitHub-Actions`? The name of a job.

# Defining the jobs

```
jobs:
  Explore-GitHub-Actions:
    runs-on: ubuntu-latest
    steps:
      - run: echo "🚀 The job was automatically triggered by a ${github.event_name} event."
```

- What is `jobs`? The key that holds all the jobs that run in the workflow file.
- What is `Explore-GitHub-Actions`? The name of a job.
- What does `runs-on: ubuntu-latest` mean?



# Defining the jobs

```
jobs:
  Explore-GitHub-Actions:
    runs-on: ubuntu-latest
    steps:
      - run: echo "🚀 The job was automatically triggered by a ${github.event_name} event."
```

- What is `jobs`? The key that holds all the jobs that run in the workflow file.
- What is `Explore-GitHub-Actions`? The name of a job.
- What does `runs-on: ubuntu-latest` mean? This workflow will run on an Ubuntu Linux virtual machine.

# Defining the jobs

```
jobs:
  Explore-GitHub-Actions:
    runs-on: ubuntu-latest
    steps:
      - run: echo "🚀 The job was automatically triggered by a ${github.event_name} event."
```

- What is `jobs`? The key that holds all the jobs that run in the workflow file.
- What is `Explore-GitHub-Actions`? The name of a job.
- What does `runs-on: ubuntu-latest` mean? This workflow will run on an Ubuntu Linux virtual machine.
- What are `steps`?

# Defining the jobs

```
jobs:
  Explore-GitHub-Actions:
    runs-on: ubuntu-latest
    steps:
      - run: echo "🚀 The job was automatically triggered by a ${github.event_name} event."
```

- What is `jobs`? The key that holds all the jobs that run in the workflow file.
- What is `Explore-GitHub-Actions`? The name of a job.
- What does `runs-on: ubuntu-latest` mean? This workflow will run on an Ubuntu Linux virtual machine.
- What are `steps`? The key that holds all of the steps to run in this job.

# The full Explore-GitHub-Actions job

YAML

```
name: GitHub Actions Demo
run-name: ${ github.actor } is testing out GitHub Actions 🚀
on: [push]
jobs:
  Explore-GitHub-Actions:
    runs-on: ubuntu-latest
    steps:
      - run: echo "🚀 The job was automatically triggered by a ${ github.event_name } event."
      - run: echo "🌍 This job is now running on a ${ runner.os } server hosted by GitHub!"
      - run: echo "🔍 The name of your branch is ${ github.ref } and your repository is ${ github.repository }."
      - name: Check out repository code
        uses: actions/checkout@v4
      - run: echo "💡 The ${ github.repository } repository has been cloned to the runner."
      - run: echo "💻 The workflow is now ready to test your code on the runner."
      - name: List files in the repository
        run: |
          ls ${ github.workspace }
      - run: echo "🟢 This job's status is ${ job.status }."
```

Explore-GitHub-Actions

succeeded 13 minutes ago in 5s

Search logs

> Set up job

> Run echo "🚀 The job was automatically triggered by a push event."

> Run echo "🌍 This job is now running on a Linux server hosted by GitHub!"

> Run echo "🔍 The name of your branch is refs/heads/octocat-patch-1 and your repository is oct..."

> Check out repository code

> Run echo "💡 The octo-org/octo-repo repository has been cloned to the runner."

> Run echo "💻 The workflow is now ready to test your code on the runner."


> List files in the repository

> Run echo "🟢 This job's status is success."


> Post Check out repository code

> Complete job

>  Run echo "🖥️ The workflow is now ready to test your code on the runner." 0s

✓  List files in the repository 0s

```
1 ▶ Run ls /home/runner/work/octo-repo/octo-repo
4 Atom
5 CONTRIBUTING.md
6 README.md
7 SUPPORT.md
8 _config.yml
9 action-a
10 issue_template.md
11 lib
12 random
13 testing-private-token-scanning.md
```

>  Run echo "🍏 This job's status is success." 0s

# How can we establish a useful set of steps?

- We can copy commands that work locally into `run`: operations
- We can find common actions on the GitHub Marketplace:  
<https://github.com/marketplace?type=actions>

# GitHub Marketplace

[Marketplace](#) / Search results

Types

Apps

**Actions** ×

Categories

API management

Chat

Code quality

Code review

Continuous integration

Dependency management

Deployment

IDEs

Learning

Localization


**Sort: Best Match** ▾

## Actions


An entirely new way to automate your development workflow.

20502 results filtered by **Actions** ×

### Actions




**First interaction**


By actions  Creator verified by GitHub

Greet new contributors when they create their first issue or open their first pull request

☆ 655 stars




**Close Stale Issues**


By actions  Creator verified by GitHub

Close issues and pull requests with no recent activity

☆ 1.1k stars




**Setup Java JDK**


By actions  Creator verified by GitHub

Set up a specific version of the Java JDK and add the command-line tools to the PATH

☆ 1.3k stars



**Setup .NET Core SDK**

By actions  Creator verified by GitHub

Used to build and publish .NET source. Set up a specific version of the .NET and authentication to private NuGet repository

☆ 826 stars

# Search for Codecov

[Marketplace](#) / Search results

Types

Apps

**Actions** ×

Categories

- API management
- Chat
- Code quality
- Code review
- Continuous integration
- Dependency management
- Deployment
- IDEs
- Learning
- Localization
- Mobile
- Monitoring
- Project management


Search:  × Sort: Best Match ▾


## Actions


An entirely new way to automate your development workflow.



12 results for "codecov" filtered by Actions ×


### Actions




**Codecov**  
By codecov  Creator verified by GitHub  
GitHub Action that uploads coverage reports for your repository to codecov.io  
☆ 1.3k stars




**Codecov ATS**  
By codecov  Creator verified by GitHub  
GitHub Action that returns selected test labels from Codecov  to CI




**Codecov reporter**  
By AlexRogalskiy  
Provides information on test coverage reports



**Codecov Action**  
By Atrox  
Sends the coverage to codecov  
☆ 8 stars



**Codecov Node**  
By iansu  
GitHub Action that uploads coverage reports to codecov.io  
☆ 3 stars



**swift-codecov**  
By mattpolzin  
Runs simple analysis on swift test codecov output  
☆ 15 stars



# Click “Use latest version”

[Marketplace](#) / [Actions](#) / Codecov



GitHub Action

Codecov

 v4.0.0-beta.3 Pre-release

Use latest version ▾

## Codecov GitHub Action

 Marketplace v3  license scan failing  Workflow for Codecov Action passing

Easily upload coverage reports to Codecov from GitHub Actions

### v4 Beta Release

v4 of the Codecov GitHub Action will use the [Codecov CLI](#) to upload coverage reports to Codecov. Currently, v4 is in beta.

#### Breaking Changes


- No current support for `arm64` and `s390x` architectures

 Verified creator

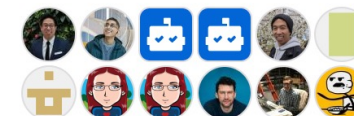
GitHub has verified that this action was created by **codecov**.

[Learn more about verified Actions.](#)

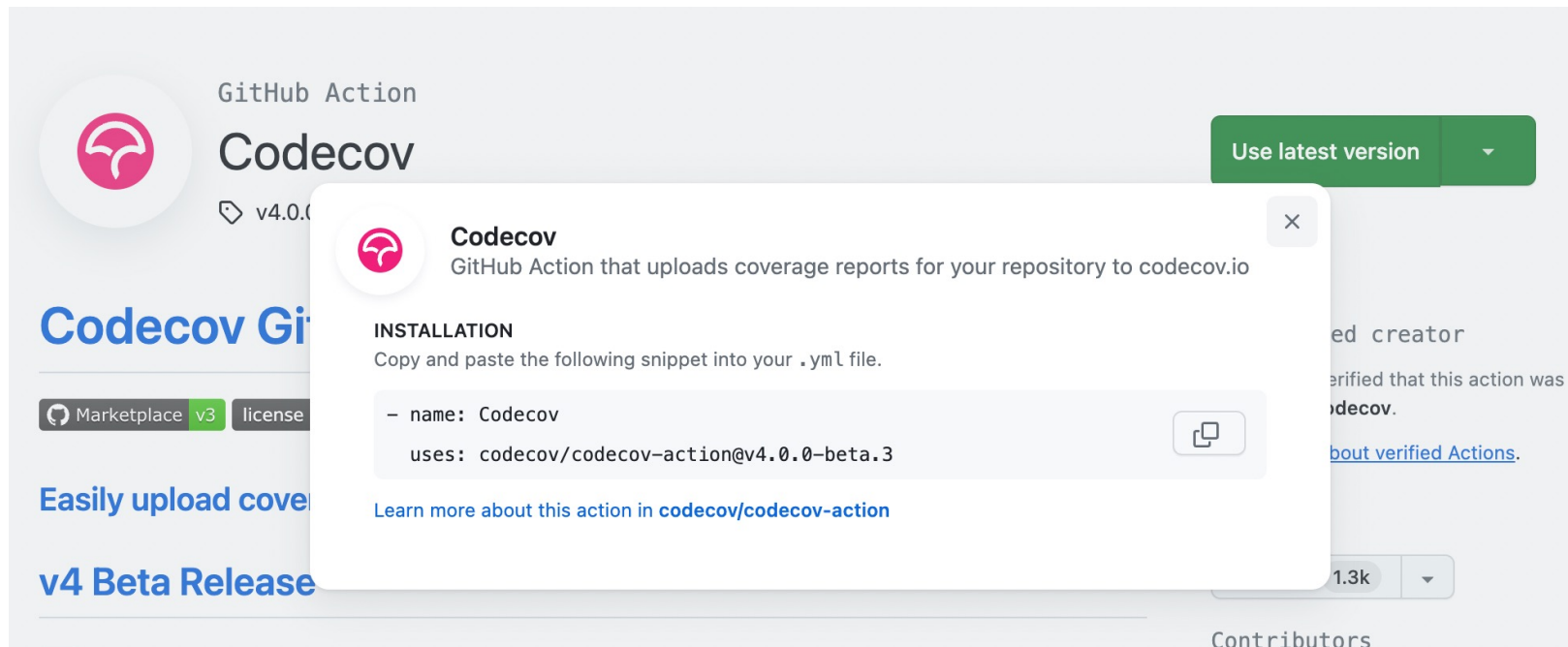
Stars

 Star 1.3k ▾

Contributors



# Copy the action as a step in your job



The screenshot shows the GitHub Action page for Codecov. The page header includes the Codecov logo, the text "GitHub Action", and the name "Codecov". A green button labeled "Use latest version" is visible. Below the header, there is a section titled "Codecov GitHub Action" with a "Marketplace v3" badge and a "license" link. A modal window is open, displaying the "INSTALLATION" instructions. The instructions state: "Copy and paste the following snippet into your .yaml file." The snippet is a YAML configuration for the Codecov action:

```
- name: Codecov
  uses: codecov/codecov-action@v4.0.0-beta.3
```

A copy icon is next to the snippet. Below the snippet, there is a link: "Learn more about this action in [codecov/codecov-action](#)". The background of the page shows a "v4 Beta Release" banner and a "Contributors" section.

# There's an action for everything



Use latest version

## Emoji action

This is a fun github action, to add random emoji to every pull request title 🚀❤️

Your github repo will look very pretty 🥰🌸🍎

📁 .github	😊 Add user config file (#6)
📄 CODE_OF_CONDUCT.md	Initial commit
📄 Dockerfile	👤 Add list of emoji to choose (#4)
📄 LICENSE	Initial commit

Stars

☆ Star 10

Contributors



Categories

Project management

Community

Links

# Exercise: Add a workflow to automatically run unit tests on `git push` to your repository

- Use `pytest`
- Can you test on multiple versions of Python? Can you test on `windows-latest` and `macos-latest`?
- Hint hint: <https://docs.github.com/en/actions/automating-builds-and-tests/building-and-testing-python>