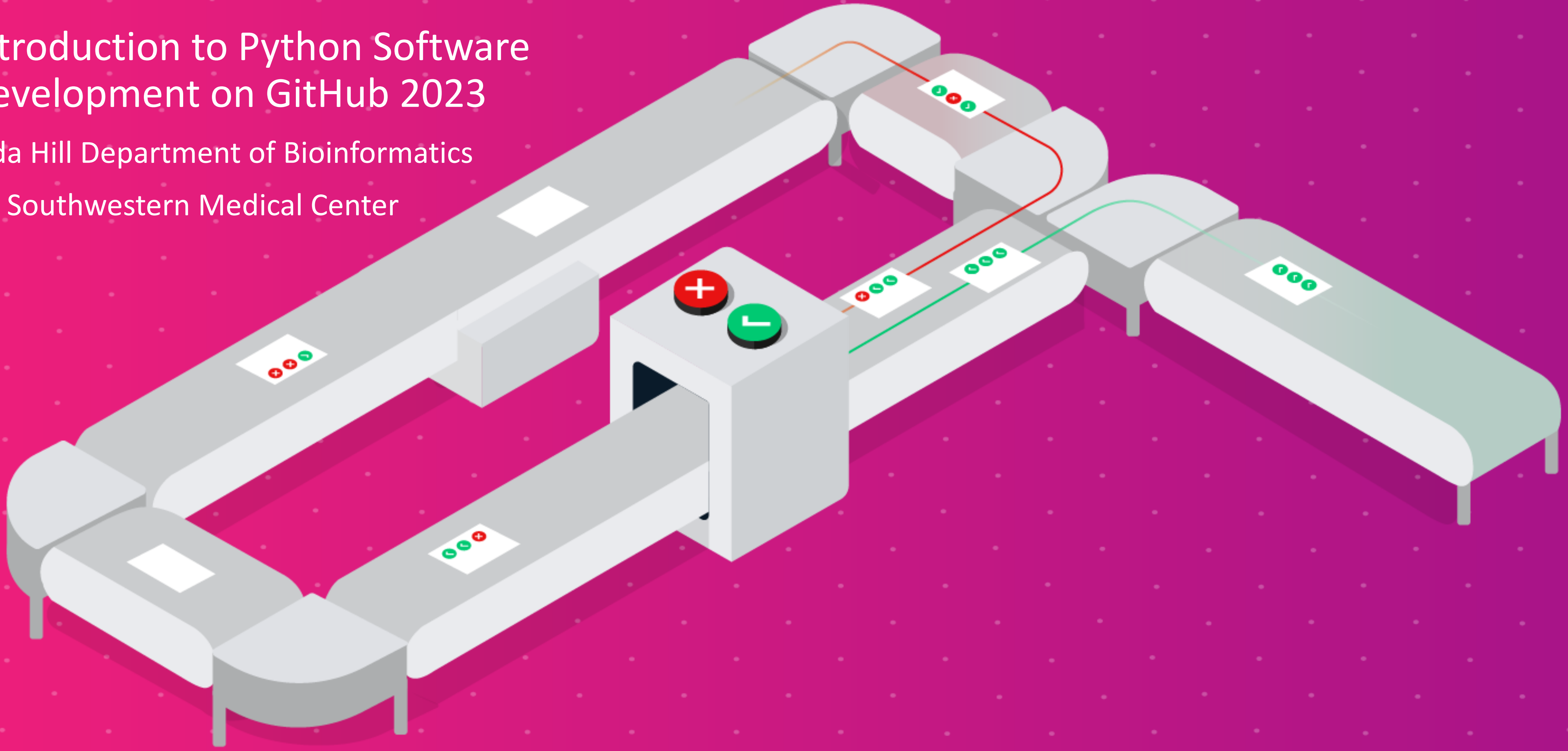
CI/CD Fundamentals

Introduction to Python Software Development on GitHub 2023

Lyda Hill Department of Bioinformatics

UT Southwestern Medical Center

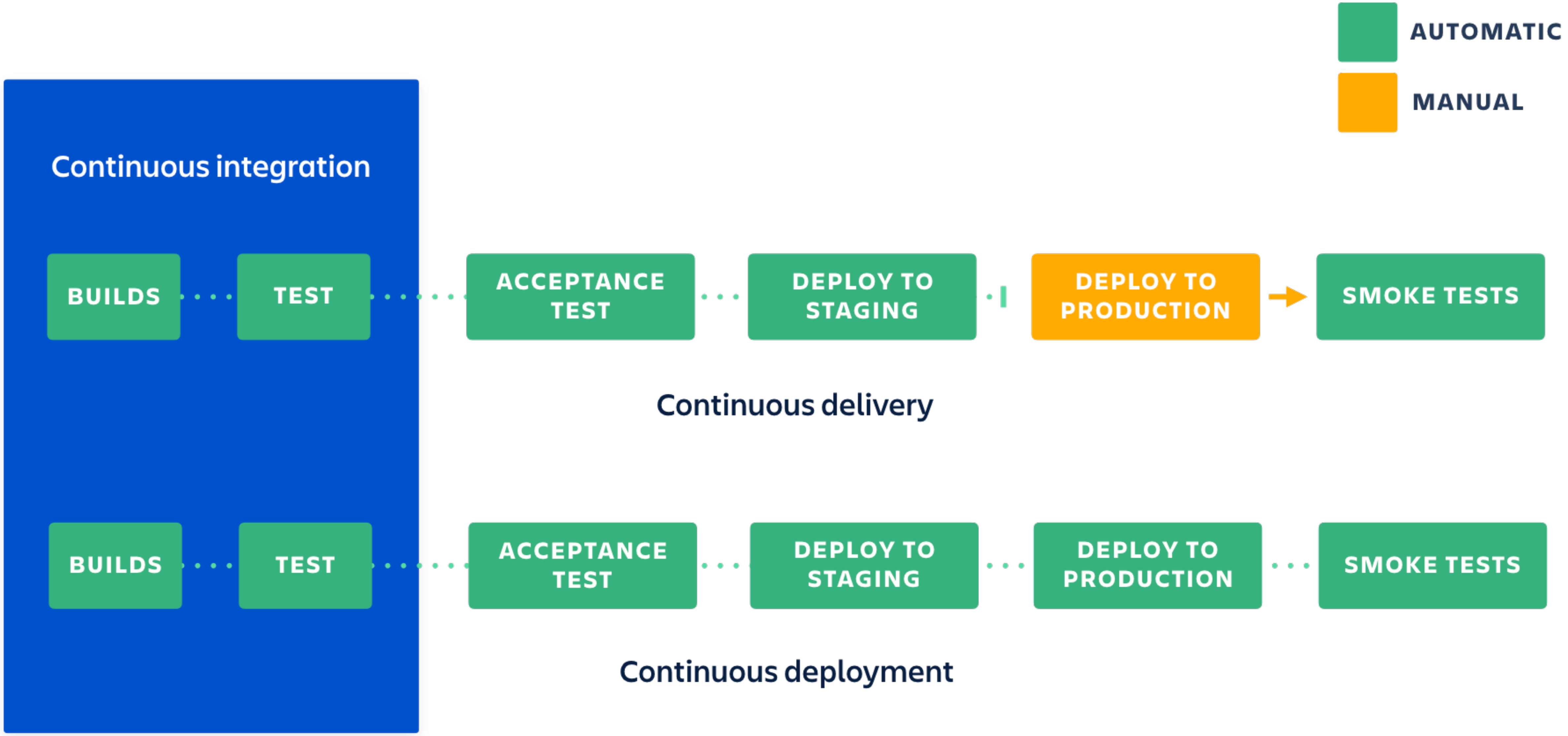


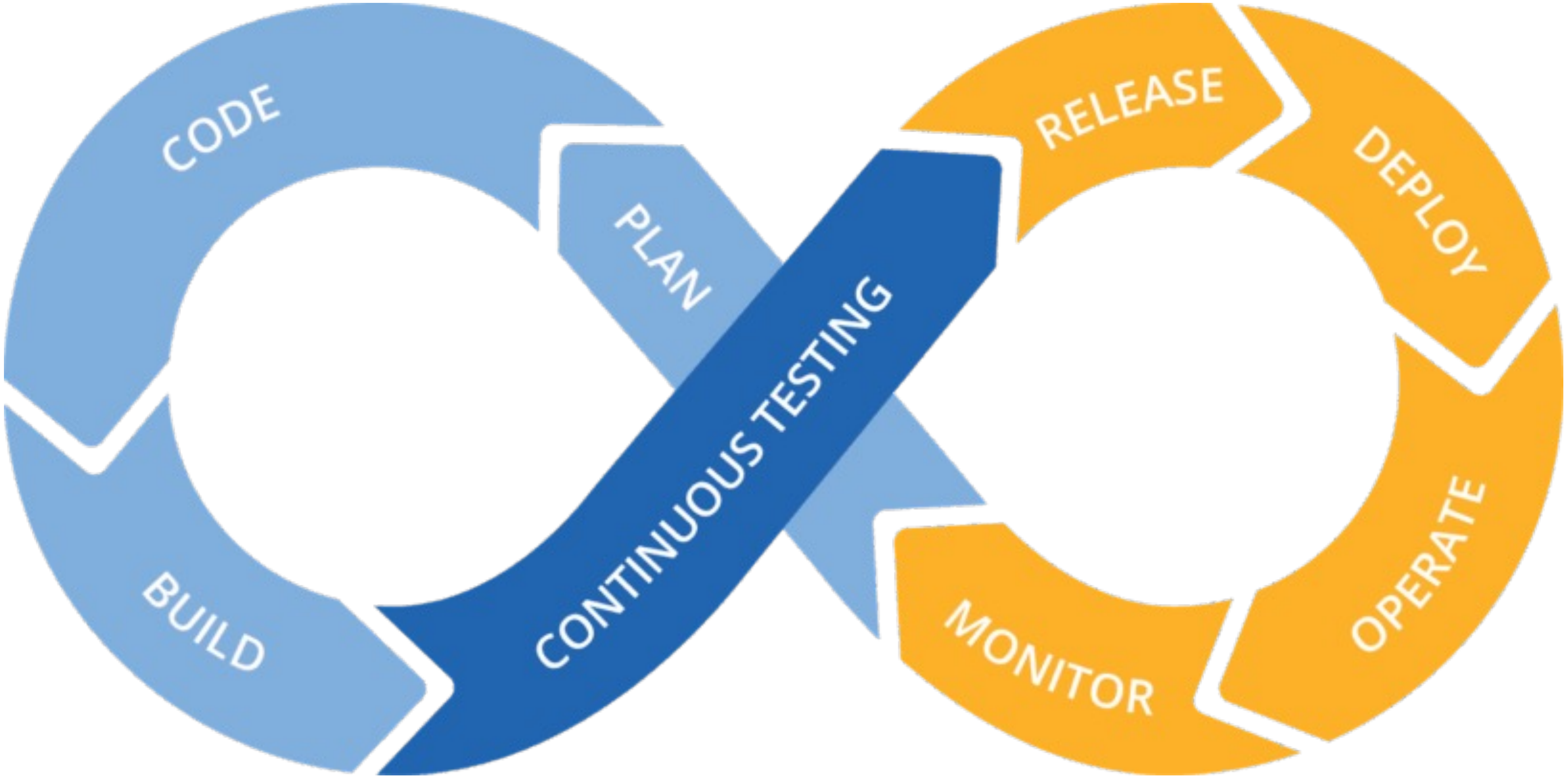
https://about.codecov.io/blog/common-steps-in-a-complete-continuous-integration-workflow/

UT Southwestern Medical Center

What does CI/CD stand for?

- **C**ontinuous **I**ntegration
- **C**ontinuous **D**elivery
- **C**ontinuous **D**eployment


AUTOMATIC
MANUAL
Continuous integration
BUILDS
TEST
ACCEPTANCE TEST
DEPLOY TO STAGING
DEPLOY TO PRODUCTION
SMOKE TESTS
Continuous delivery
BUILDS
TEST
ACCEPTANCE TEST
DEPLOY TO STAGING
DEPLOY TO PRODUCTION
SMOKE TESTS
Continuous deployment



https://deploybot.com/blog/continuous-development

Vocabulary

- **Build**

Vocabulary

- **Build** – Convert source code files into a standalone software that anyone can run on their machine.

Vocabulary

- **Build** – Convert source code files into a standalone software that anyone can run on their machine.

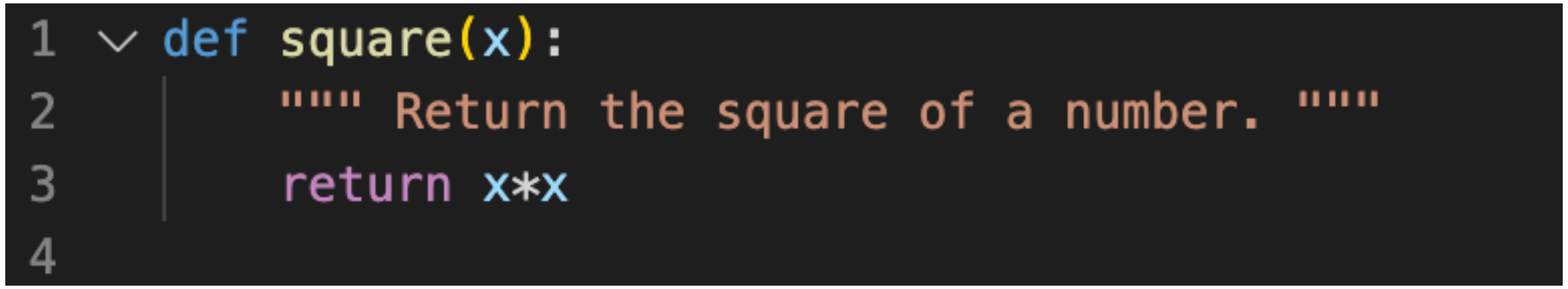
- **Test**

Vocabulary

- **Build** – Convert source code files into a standalone software that anyone can run on their machine.

- **Test** – Evaluate and verify software can do what it is supposed to do.

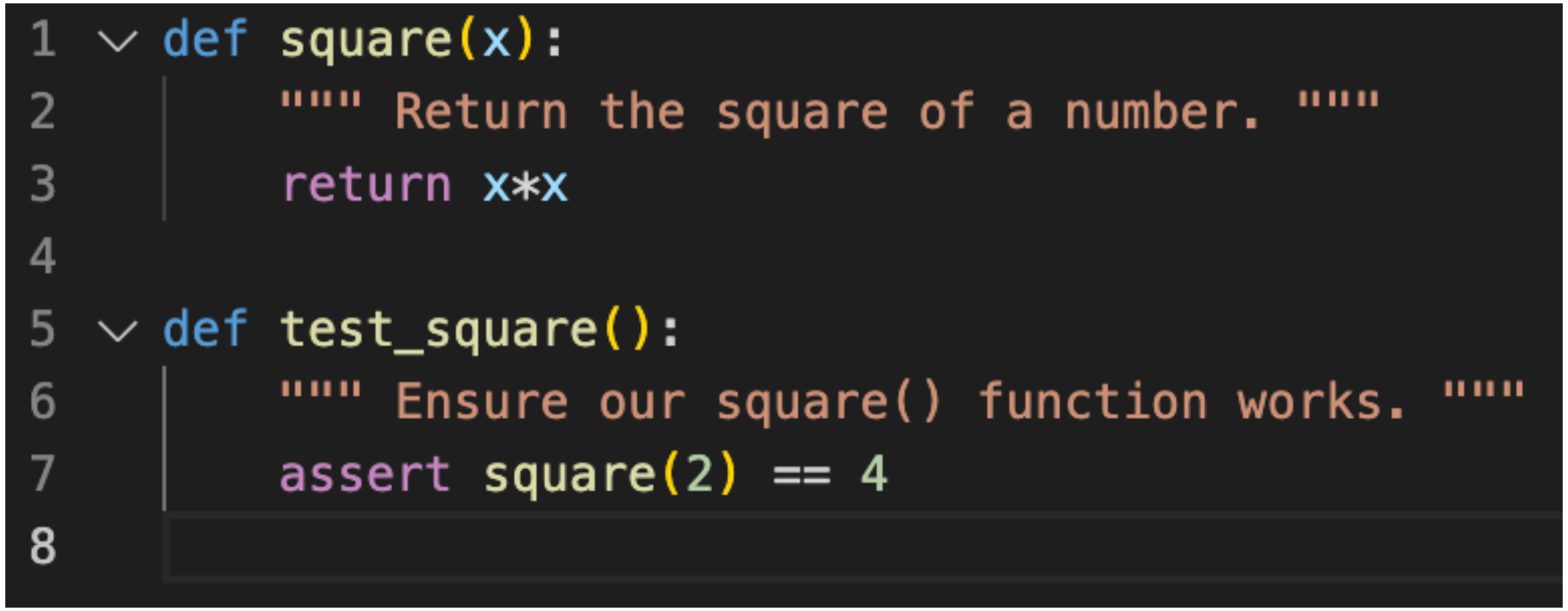
How do we know this function works?



```python
def square(x):
    """ Return the square of a number. """
    return x**x

```

We check it with a unit test



```python
def square(x):
    """ Return the square of a number. """
    return x**x

def test_square():
    """ Ensure our square() function works. """
    assert square(2) == 4

```

Test subgroups

- **Acceptance test** – A group of unit tests that ensure the software meets specifications e.g. of a contract.

- **Smoke test** – A group of unit tests that act as a sanity check for severe failures. If you run the software, does smoke come out of the computer?

Vocabulary

- **Build** – Convert source code files into a standalone software that anyone can run on their machine.
- **Test** – Evaluate and verify software can do what it is supposed to do.
- **Release**

Vocabulary

- **Build** – Convert source code files into a standalone software that anyone can run on their machine.
- **Test** – Evaluate and verify software can do what it is supposed to do.
- **Release** – A build that is a new or upgraded version of the software.

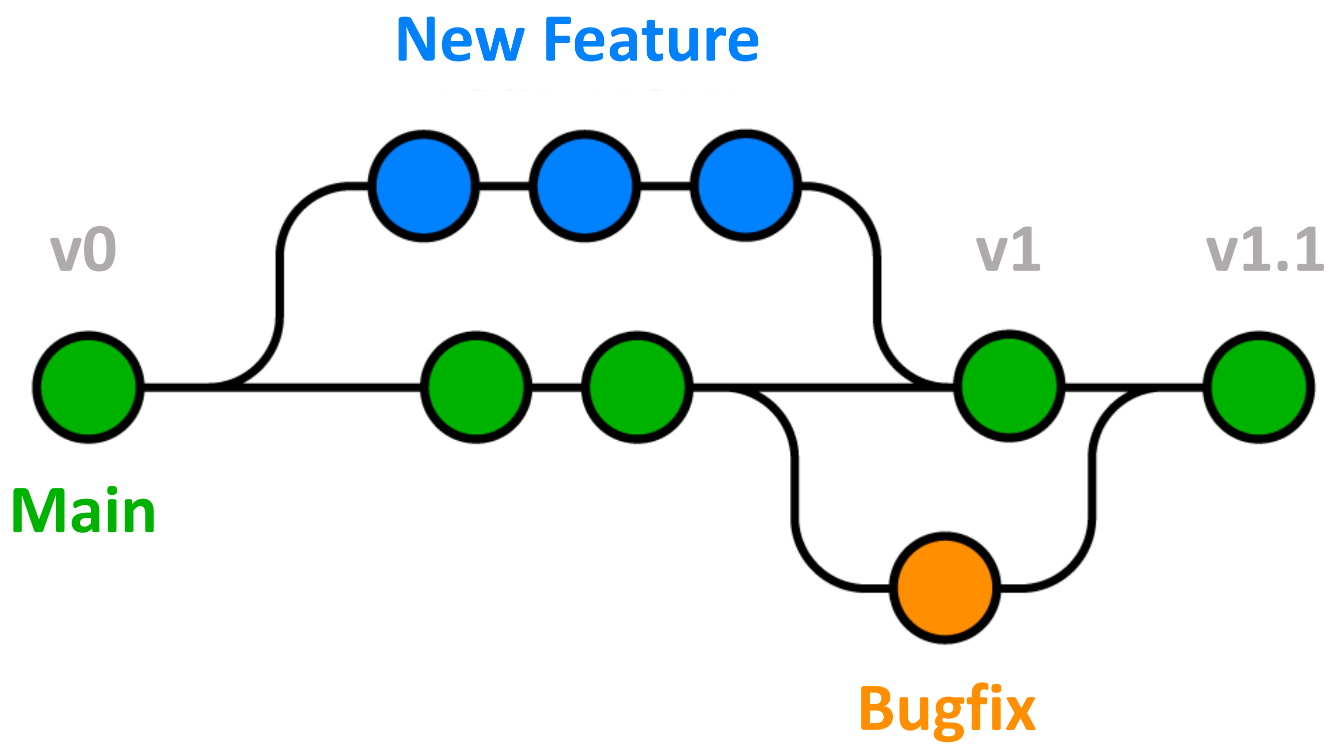
Vocabulary

- **Build –** Convert source code files into a standalone software that anyone can run on their machine.
- **Test** – Evaluate and verify software can do what it is supposed to do.
- **Release** – A build that is a new or upgraded version of the software.
- **Deploy**

Vocabulary

- **Build –** Convert source code files into a standalone software that anyone can run on their machine.
- **Test** – Evaluate and verify software can do what it is supposed to do.
- **Release** – A build that is a new or upgraded version of the software.
- **Deploy** – Make the software available for use.

The goal of CI/CD is to ensure developments do not stray far from the main branch



Why is CI/CD useful?

- Ensures disparate parts of the code base work together throughout development, preventing integration challenges.

- Protects against release of broken software.

- Allows for fast feedback from users and fast fixes from developers.

How do we implement CI/CD in practice?

- **Version control** (`git`)
- **Automatic testing** (`pytest`)
- **Automatic building** (`setuptools`, `pyproject.toml`)
- **Automatic deployment** (`twine`, `PyPI`)

Additional CI/CD tools in the workflow

- **Code quality** (linter such as `ruff`, code formatter such as `black`)
- **Test coverage check** (`codecov`)
- **Documentation** (`sphinx`, `numpydoc`)
- **Security checks** (`CodeQL`)

Local CI/CD workflows

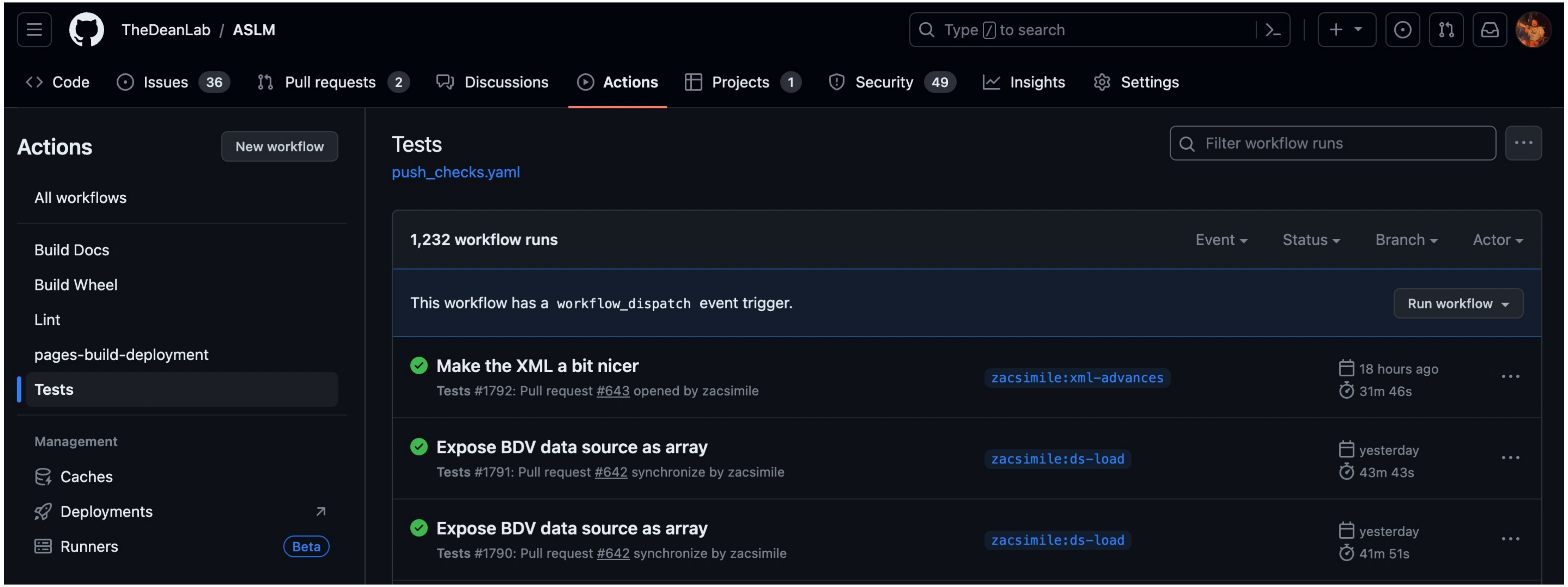
- Can run some tools, such as the linter and code formatter, before pushing code to the repository

- Can automatically run some actions using `pre-commit`

Running the CI/CD workflow

- Need a continuous integration tool
  - Bitbucket (https://bitbucket.org/product/features/pipelines)
  - Jenkins (https://jenkins.io)
  - AWS CodePipeline (https://aws.amazon.com/codepipeline)
  - CircleCI (https://circleci.com)
  - Azure (https://azure.microsoft.com/)
  - Gitlab (https://about.gitlab.com/)
  - GitHub (https://github.com/)
  - Etc.
- These tools use a YAML file (or similar) to describe a series of actions that make up a workflow.

GitHub Actions Dashboard



TheDeanLab / ASLM

Type / to search

Code | Issues 36 | Pull requests 2 | Discussions | Actions | Projects 1 | Security 49 | Insights | Settings

Actions — New workflow

- All workflows
- Build Docs
- Build Wheel
- Lint
- pages-build-deployment
- Tests

Management

- Caches
- Deployments
- Runners Beta

Tests
push_checks.yaml

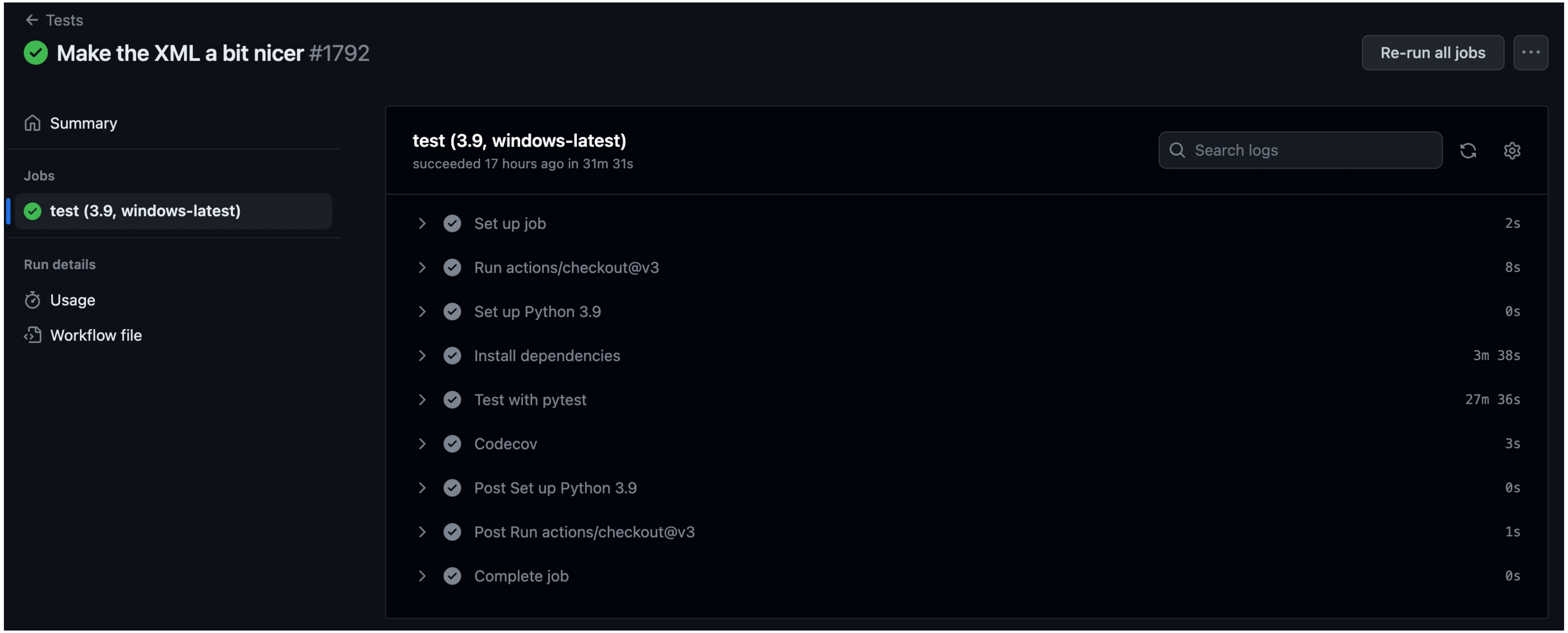
Filter workflow runs

1,232 workflow runs — Event, Status, Branch, Actor

This workflow has a `workflow_dispatch` event trigger. Run workflow

Run	Branch	When	Duration
Make the XML a bit nicer Tests #1792: Pull request #643 opened by zacsimile	zacsimile:xml-advances	18 hours ago	31m 46s
Expose BDV data source as array Tests #1791: Pull request #642 synchronize by zacsimile	zacsimile:ds-load	yesterday	43m 43s
Expose BDV data source as array Tests #1790: Pull request #642 synchronize by zacsimile	zacsimile:ds-load	yesterday	41m 51s

GitHub Actions Workflow Example



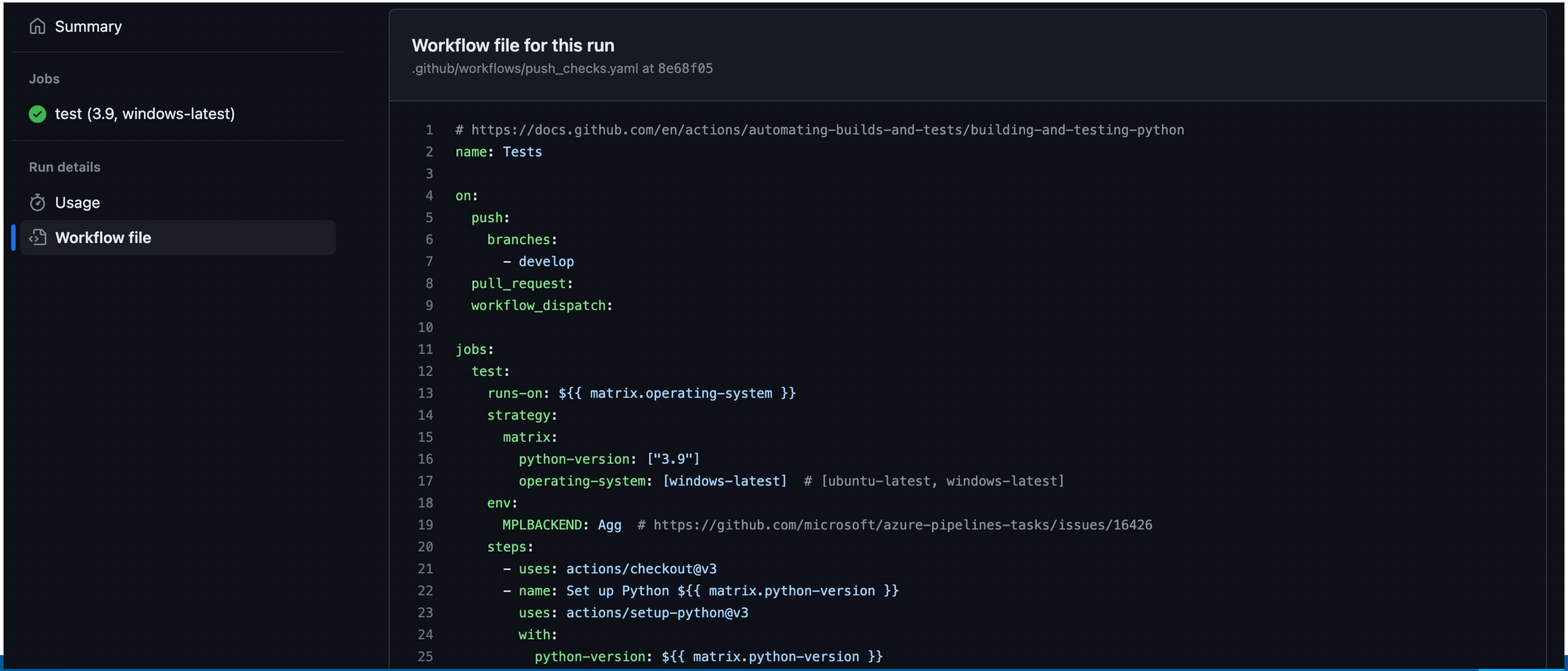
Make the XML a bit nicer #1792

test (3.9, windows-latest)

test (3.9, windows-latest)
succeeded 17 hours ago in 31m 31s

```
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



Summary

Jobs

test (3.9, windows-latest)

Run details

Usage

Workflow file

Workflow file for this run

.github/workflows/push_checks.yaml at 8e68f05

```yaml
# https://docs.github.com/en/actions/automating-builds-and-tests/building-and-testing-python
name: Tests

on:
  push:
    branches:
      - develop
  pull_request:
  workflow_dispatch:

jobs:
  test:
    runs-on: ${{ matrix.operating-system }}
    strategy:
      matrix:
        python-version: ["3.9"]
        operating-system: [windows-latest]  # [ubuntu-latest, windows-latest]
    env:
      MPLBACKEND: Agg  # https://github.com/microsoft/azure-pipelines-tasks/issues/16426
    steps:
      - uses: actions/checkout@v3
      - name: Set up Python ${{ matrix.python-version }}
        uses: actions/setup-python@v3
        with:
          python-version: ${{ matrix.python-version }}
```