

Continuous Integration and Continuous Delivery (CI/CD)

Software Engineering Summer School
June 2025

Amy Defnet

What is CI/CD?

- Continuous Integration (CI)
 - Frequent merging of changes into main branch with active quality verification
- Continuous Deployment (CD)
 - Automatic release of new software versions

What is CI for?

- Building documentation
 - Run Sphinx/Doxygen/etc to generate documentation in pdf or html form
 - Check if examples in the documentation work
- Build static websites
 - Build and publish a website to GitHub Pages/ReadTheDocs/etc
- Generating pull requests for updates
 - Update dependencies
 - Fix typos and coding style
- Whatever your project needs!

What are the benefits?

- Consistent, controlled environment between runs
- Runs every PR/commit/tag/whatever you choose
- Can't be skipped or forgotten, no contributor setup
- Can run lots of OS's, Python versions, compilers, etc.

Some major CI services

- **Travis CI**: Very popular for years, but not anymore.
- **Jenkins**: A self-host only OSS solution.
- **Circle CI**: The first more “modern” design.
- **GitLab CI**: For years, this was one of the best services. Still very good.
- **Azure Pipelines**: Very modular design is easy to upgrade and maintain.
- **GitHub Actions (GHA)**: Extremely simple and popular. Actions are easy to write and share.

Today we will be focusing on GitHub Actions

We first need to briefly discuss exit codes and YAML

Exit Codes

- Every time you run a command in a shell there is an exit code that indicates if it ran successfully, or if there was an error.
- An exit code of 0 indicates that the command ran successfully, other numbers indicate an error.
- Sometimes different numbers correspond to different errors.

Exit codes

```
> mkdir test  
> echo $?  
0
```

```
> mkdir test  
mkdir: test: File exists  
> echo $?  
1
```

```
> mkdir -z test  
mkdir: illegal option - z  
usage: mkdir [-pv] [-m mode] directory_name ...  
> echo $?  
64
```


Exit Codes

- CI workflows will typically stop once they encounter a non-zero exit code.
- Sometimes you may need to run a command that might fail, but you want the workflow to proceed.
- Some scripts and binaries don't respect this standard and return non-zero exit codes even when successful.

Exit Codes

- Error codes can be ignored using logical Or (||)

```
> mkdir -z test || echo "ignore"  
ignore
```

YAML

YAML (YAML Ain't Markup Language, originally Yet Another Markup Language) is a human-readable data serialization language.

- Easy to read and use
- Very commonly used in CI configuration files
- File extension is .yaml or .yml

YAML: Defining Scalar Values

```
number-value: 42
boolean-value: true # can also be on or yes
string-value: "Hello world"
another-string: String without quotes
```

YAML: Defining lists

```
colors:  
  - red  
  - green  
  - blue
```

```
more_colors: [black, white]
```

YAML: Defining dictionaries

```
person:  
  name: John Smith  
  age: 33  
  occupation: accountant  
  
same_person: {name: John Smith, age: 33, occupation: accountant}
```

YAML: Defining multi-line strings

```
some-text: >
  Multiple
  lines
  of
  text
same-text: "Multiple lines of text\n"
```

YAML: Defining multi-line strings

```
some-text: |  
  Multiple  
  lines  
  of  
  text  
same-text: "Multiple\nlines\nof\ntext\n"
```


Setting up GitHub Workflows

- Each workflow is configured by a yaml file placed in `.github/workflows`
- Can be set to trigger by a wide variety of events
- Can run your own commands or use actions written by you or third-parties

Basic Structure of Workflow File

```
on: <event or list of events>

jobs:
  job_1:
    name: <name of job>
    runs-on: <type of machine>
    steps:
      - uses: <some third-party action>
      - name: <name of step>
        run: <command>

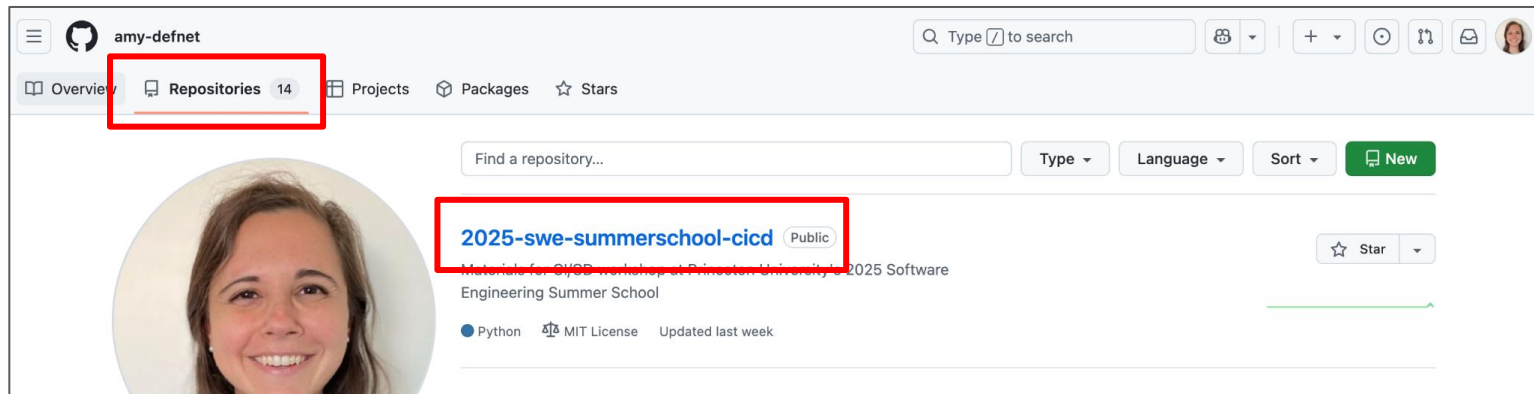
  job_2:
    runs-on: <type of machine>
    steps:
      - uses: <command>
```

Hands-On Exercises

Go to the following link:

<https://github.com/amy-defnet/2025-swe-summer-school-cicd>

Or <https://github.com/amy-defnet> → Repositories



github.com/amy-defnet/2025-swe-summer-school-cicd

amy-defnet / 2025-swe-summer-school-cicd

Type to search

<> Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

Settings

2025-swe-summer-school-cicd

Public template

Pin

Watch 0

Fork 0

Star 0

Use this template

main

1 Branch

0 Tags

Go to file

t

Add file

<> Code

amy-defnet

update README

190151b · 1 minute ago

4 Commits

src/cicdexample	Initial commit	2 weeks ago
tests	Initial commit	2 weeks ago
.gitignore	Initial commit	2 weeks ago
LICENSE	Initial commit	2 weeks ago
README.md	update README	1 minute ago
pyproject.toml	Initial commit	2 weeks ago

About

Materials for CI/CD workshop at Princeton University's 2025 Software Engineering Summer School

Readme

MIT license

Activity

0 stars

0 watching

0 forks

Releases

Exercise: Hello CI World

```
name: Example
on: push
jobs:
  greeting:
    runs-on: ubuntu-latest
    steps:
      - name: Greeting!
        run: echo hello world
```

Exercise: CI for Python Testing


```
name: Code Checks
on: push
jobs:
  greeting:
    runs-on: ubuntu-latest
    steps:
      - name: Greeting!
        run: echo hello world

  test-python-3-10:
    name: Check Python 3.10 on Ubuntu
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v3

      - uses: actions/setup-python@v4
        with:
          python-version: "3.10"

      - name: Install package
        run: python -m pip install -e .[test]

      - name: Test package
        run: python -m pytest
```

Exercise: Python Matrix

```
name: Code Checks
on: push
jobs:

  test-python-versions:
    name: Check Python ${ matrix.python-version } on Ubuntu
    runs-on: ubuntu-latest
    strategy:
      matrix:
        python-version: ["3.10", "3.11"]
    steps:
      - uses: actions/checkout@v3

      - name: Setup Python ${ matrix.python-version }
        uses: actions/setup-python@v4
        with:
          python-version: ${ matrix.python-version }

      - name: Install package
        run: python -m pip install -e .[test]

      - name: Test package
        run: python -m pytest
```

Helpful Resources

- [GitHub Actions Documentation](#)
- [Weekly Research Computing Help Sessions](#)
 - Mondays 2:30-4:30pm - Stokes Library, room 070
 - Wednesdays 1:00-3:00pm - Fine Visualization Library
 - Thursdays 1:00-3:00pm - Fine Visualization Library