

Continuous Integration

Or

Let the Damn Computer Worry About It

Jacob Seiler

State of the Game

- Have a repo which contains tests run using the command ‘pytest’.

State of the Game

- Have a repo which contains tests run using the command ‘pytest’.
- Ensures your code can run on your machine!

State of the Game

- Have a repo which contains tests run using the command ‘pytest’.
- Ensures your code can run on your machine!
- Make sure you run ‘pytest’ before you push a commit!

I'm Lazy and Don't Like Thinking

- Remembering to run PyTest **every single time** before you push is an absolute nightmare.

I'm Lazy and Don't Like Thinking

- Remembering to run PyTest **every single time** before you push is an absolute nightmare.

I'm Lazy and Don't Like Thinking

- Remembering to run PyTest **every single time** before you push is an absolute nightmare.
- What happens if you forget and your code is sitting on GitHub broke...?

I'm Lazy and Don't Like Thinking

- Remembering to run PyTest **every single time** before you push is an absolute nightmare.
- What happens if you forget and your code is sitting on GitHub broke...?
- Also, the tests run only on your machine. What happens if someone is running Windows? Or a different Python version?

I'm Lazy and Don't Like Thinking

- Remembering to run PyTest **every single time** before you push is an absolute nightmare.
- What happens if you forget and your code is sitting on GitHub broke...?
- Also, the tests run only on your machine. What happens if someone is running Windows? Or a different Python version?
- What if we set it up so our tests automatically run every time we push? And what if we could decide what environment to run these tests on...?

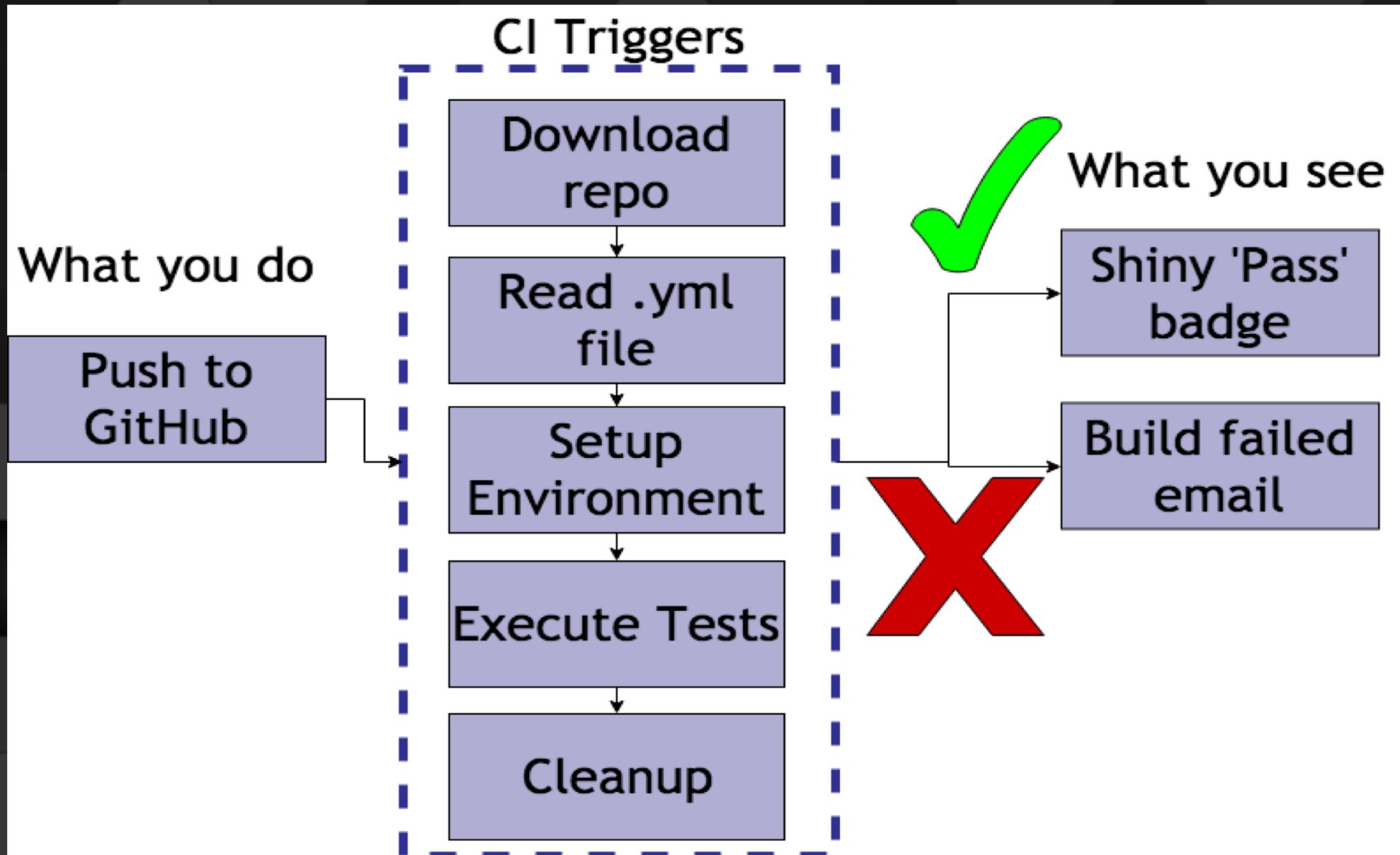
I'm Lazy and Don't Like Testing

- Remembering to run `npm test` at absolute nightime

Continuous Integration

- We automatically run every time we push tests

Continuous Integration



The .travis.yml File

- What Python versions or OS should be tested?

The .travis.yml File

- What Python versions or OS should be tested?
- What should Travis do once it's cloned your repo?

The .travis.yml File

- What Python versions or OS should be tested?
- What should Travis do once it's cloned your repo?
- What happens if the tests pass/fail?

The .travis.yml File

- What Python versions or OS should be tested?
- What should Travis do once it's cloned your repo?
- What happens if the tests pass/fail?
- What happens after your tests have been executed?

Your Turn!

- Log into Travis CI and enable builds for your fork of the workshop repo.

Your Turn!

- Log into Travis CI and enable builds for your fork of the workshop repo.
- Take the template .yml file and adjust it to be ‘correct’.

Your Turn!

- Log into Travis CI and enable builds for your fork of the workshop repo.
- Take the template .yml file and adjust it to be ‘correct’.
- Track the .yml file. Commit. Push!

Your Turn!

- Log into Travis CI and enable builds for your fork of the workshop repo.
- Take the template .yml file and adjust it to be ‘correct’.
- Track the .yml file. Commit. Push!
- Did your tests pass?
 - No? Change the .yml file and commit/push until it does!
 - Yes? Add the shiny ‘pass’ badge to the README!

Your Turn!

- Log into Travis CI and enable builds for your fork of the workshop repo.
- Take the template .yml file and adjust it to be ‘correct’.
- Track the .yml file. Commit. Push!
- Did your tests pass?
 - No? Change the .yml file and commit/push until it does!
 - Yes? Add the shiny ‘pass’ badge to the README!
- All done? Work on getting ‘codecov’ to automatically run and update your results to ‘codecov.io’. Then get the codecov badge!