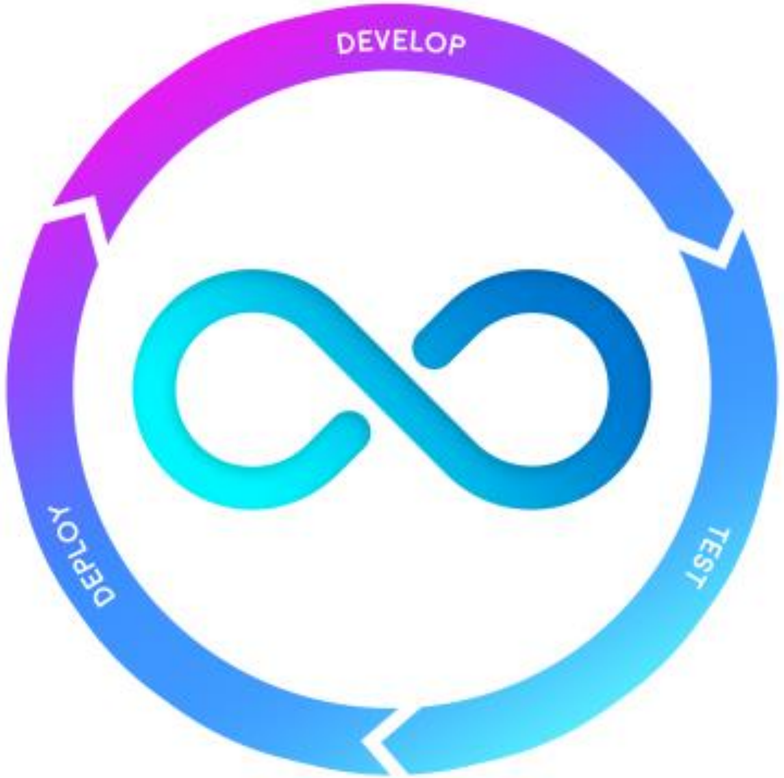
PART I OF **PRODUCTION READY CODE SERIES**



Continuous **Integration**

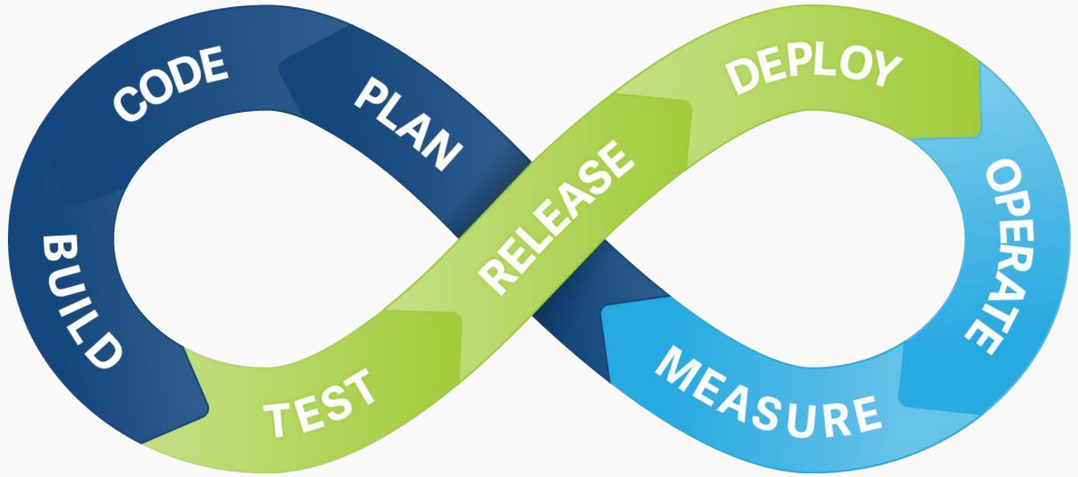
PRESENTED BY CSEC

Table of Contents

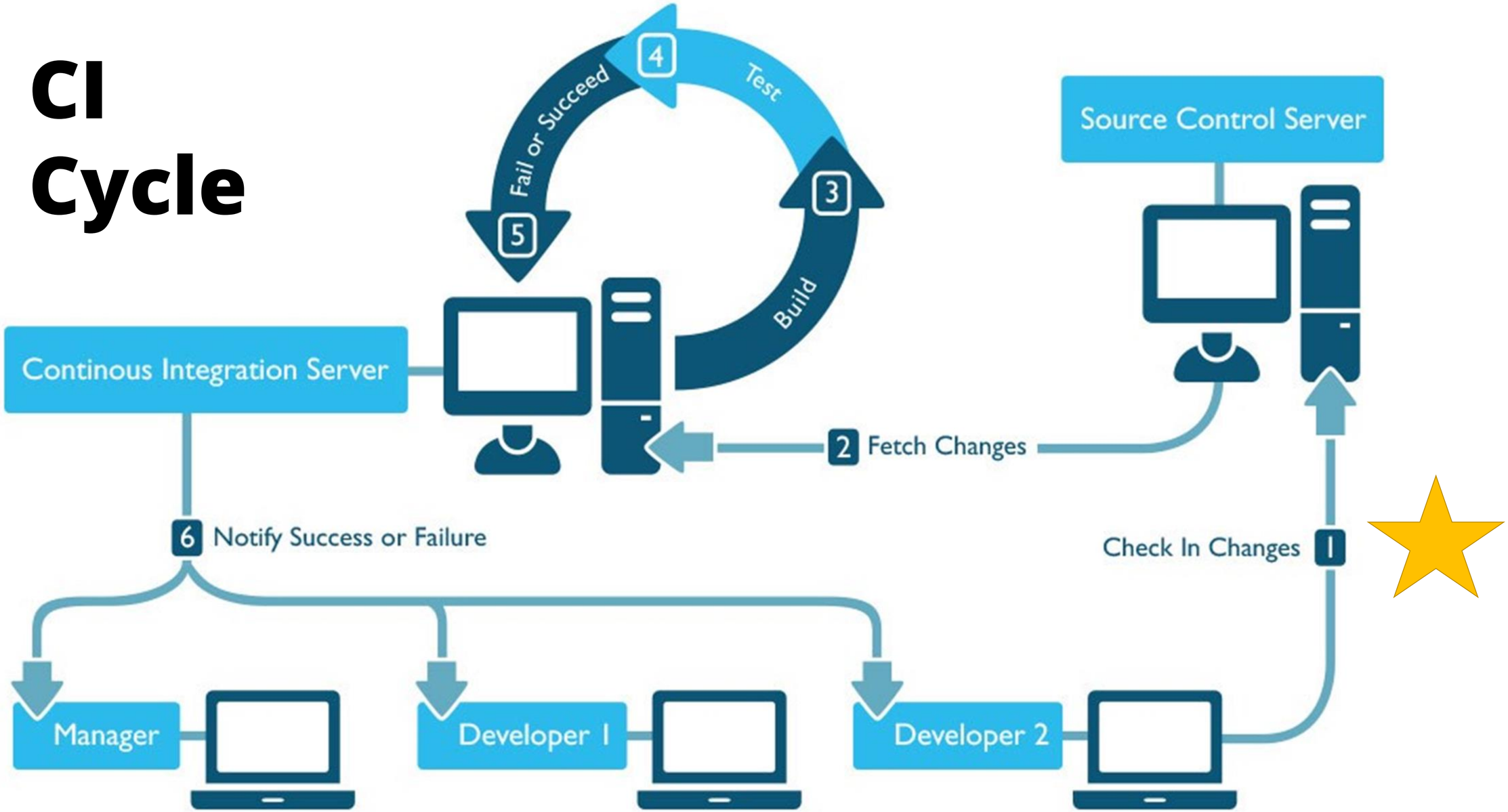
What is Continuous Integration?

Continuous Integration (CI) is the practice of routinely integrating code changes into the main or development branches of a repository, and testing the changes, as early and often as possible.

Ideally, developers will integrate their code daily, if not multiple times a day.



CI Cycle

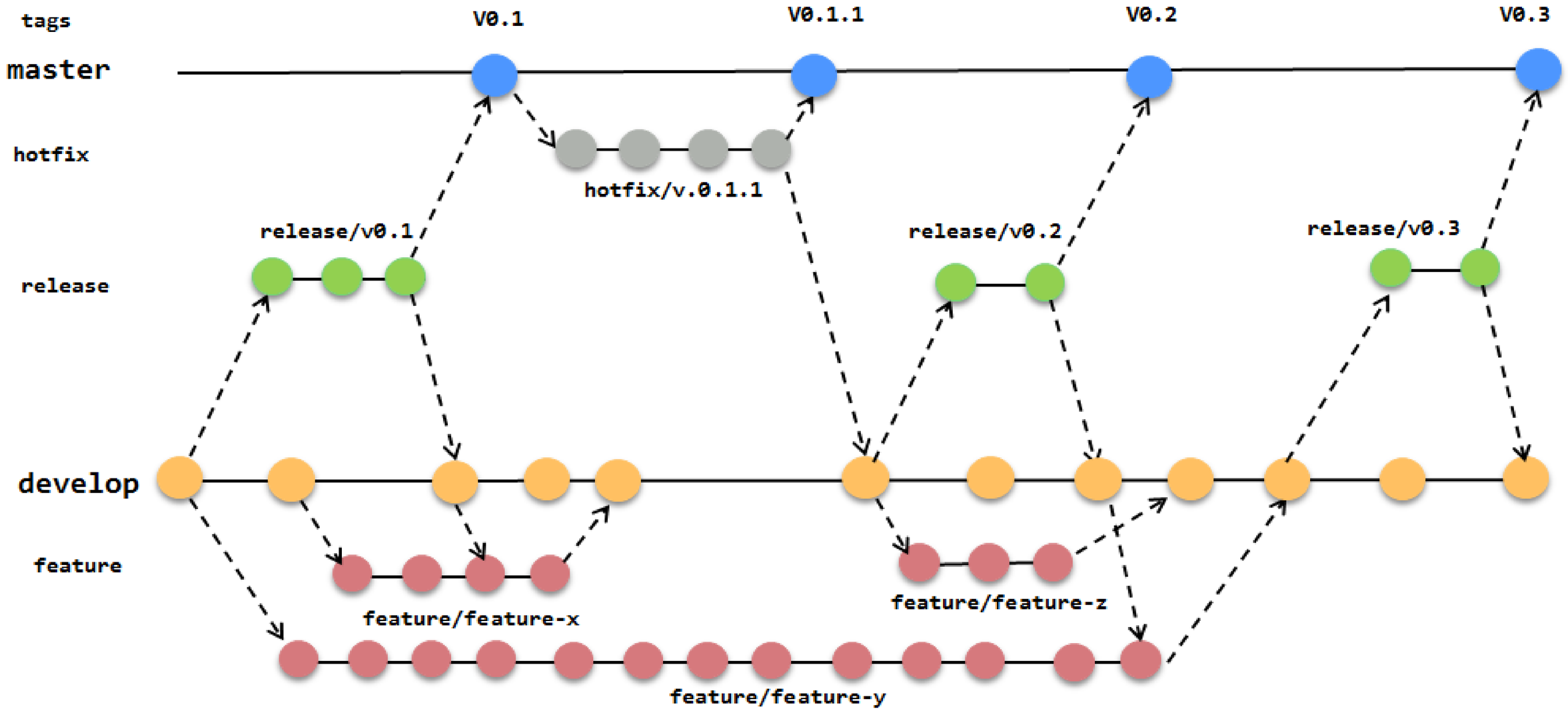

Source Control Server
Continous Integration Server
Fail or Succeed
4
Test
3
Build
5
2 Fetch Changes
6 Notify Success or Failure
Check In Changes 1
Manager
Developer 1
Developer 2

Benefits of Continuous Integration?

- CI offers a real-time window into the actual state of the software system and associated quality measurements, allowing immediate and constant engagement of all team members, including operations, product owners and QA, throughout the project lifecycle.

- CI is also a great way to reduce the amount of time spent on fixing bugs and regression by putting more emphasis on having a good test suite.

Project Example




MERGE
HELL

The CI Landscape

- Currently there are many popular tools being used in the CI space such as Travis CI, Jenkins, Circle CI, Gitlab CI to name a few.

- Some are intuitive, easy to setup and have excellent integrations with GitHub such as Travis CI (popular among open source communities)

- Some are highly customizable and on-perm like Jenkins, allowing you to control every aspect of your DevOps pipeline in house.

Example in a Scenario

Jin is a Software Engineering intern at a big eCommerce company.

He is tasked with a user story for a **new feature** request for an internal microservice. Being a UTSC student, he's obviously brilliant so he finishes creating the feature early and creates a pull request for his branch to be merged to the master.

Jin claims **he created and ran all the tests** on his computer and his description follows all the guidelines, so Rachel, being the senior engineer approves his pull request.

Example in a Scenario

A few days later, it is noticed that the company's APIs start intermittently failing and everyone is firefighting to patch the issue.

Eventually the faulty feature that caused all downstream microservices to fail is identified, but by that time the company has lost a million dollars in revenue.

How could have CI helped prevent this tragic situation?

Alex Li

Apparently my coworker intern already cost the company $100,000 due to bug

😄😢 2

Intern season

kuroba

Ayy

At wish?

Do you have a testing framework for your diffs at wish?

yeah what kinda stuff was it that the testing didn't catch

kuroba

That too

But the funny thing is

Each test run usually costs like 30$

For a company at the scale of wish

aws instances?

kuroba

Yeah, usually EC2

At fb, each test run was about 80$

Alex Li

Apparently it was a logic error on one line

I dont know full context

wish

Day 8

Take a break and open your laptops!

matplotlib: plotting with Python http://matplotlib.org/

27,967 commits | 8 branches | 76 releases | 772 contributors

Branch: master | New pull request | Create new file | Upload files | Find file | Clone or download

README.rst

Matplotlib

Matplotlib is a Python 2D plotting library which produces publication-quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shell (à la MATLAB or Mathematica), web application servers, and various graphical user interface toolkits.

NOTE: The current master branch is now Python 3 only. Python 2 support is being dropped.

Home page

Installation

For installation instructions and requirements, see the INSTALL.rst file or the install documentation. If you think you may want to contribute to matplotlib, check out the guide to working with the source code.

matplotlib / matplotlib

Watch 527 | Star 8,249 | Fork 3,802

Code | Issues 1,174 | Pull requests 291 | Projects 5 | Wiki | Insights

matplotlib: plotting with Python http://matplotlib.org/

27,967 commits | 8 branches | 76 releases | 772 contributors

Branch: master | New pull request | Create new file | Upload files | Find file | Clone or download

README.rst

Matplotlib

Matplotlib is a Python 2D plotting library which produces publication-quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shell (à la MATLAB or Mathematica), web application servers, and various graphical user interface toolkits.

NOTE: The current master branch is now Python 3 only. Python 2 support is being dropped.

Home page

Installation

For installation instructions and requirements, see the INSTALL.rst file or the install documentation. If you think you may want to contribute to matplotlib, check out the guide to working with the source code.

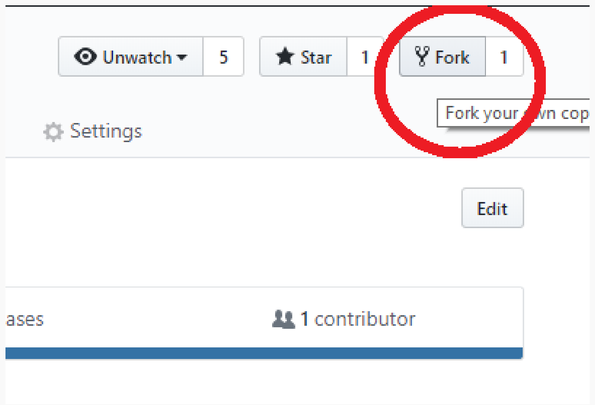
Working with CI Workshop

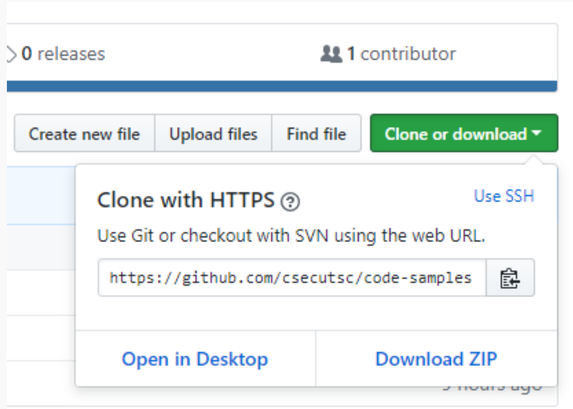
- Forking and Cloning a repository
- Creating **GitHub integrations** w/TravisCI and Coveralls
- Defining **.travis.yml**
  - Running environment and OS
  - Code styles and linting
  - Running tests
  - Coverage reports
- Reading the Travis CI report
- Analyzing additional tools
- Adding the buttons to your GitHub repository

Forking and Cloning a Repo

- We'll be working with a repo that has a Palindrome checker

github.com/csecutsc/code-samples





Read the Code and Understand

Two main files:

palindromes.py

the logic behind palindrome detection and processing

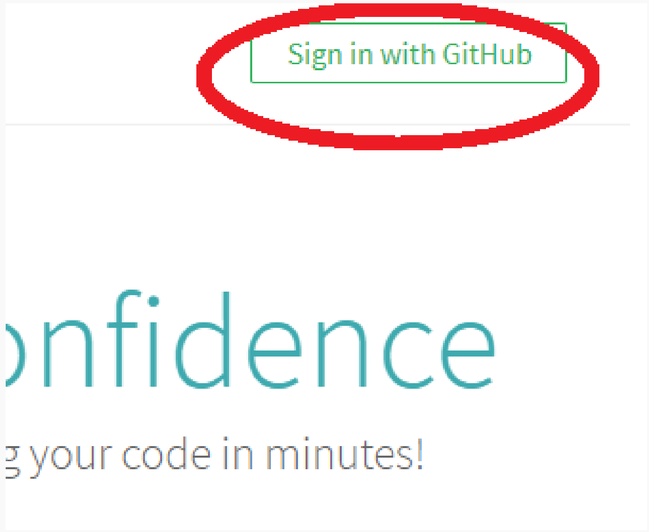
test-palindromes.py

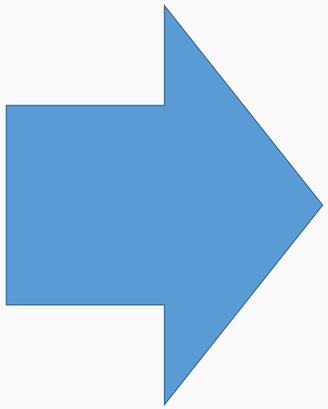
the test suite that executes tests for certain functions in palindromes.py

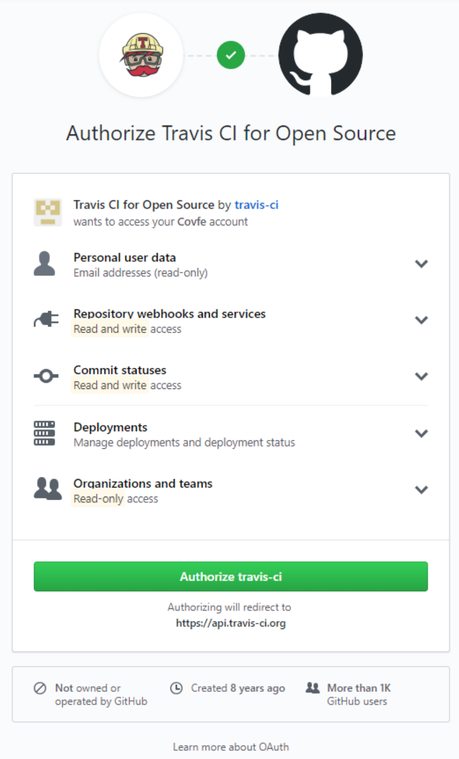
Setting up Travis CI

- Navigate to Travis CI website and click **Sign in with GitHub**

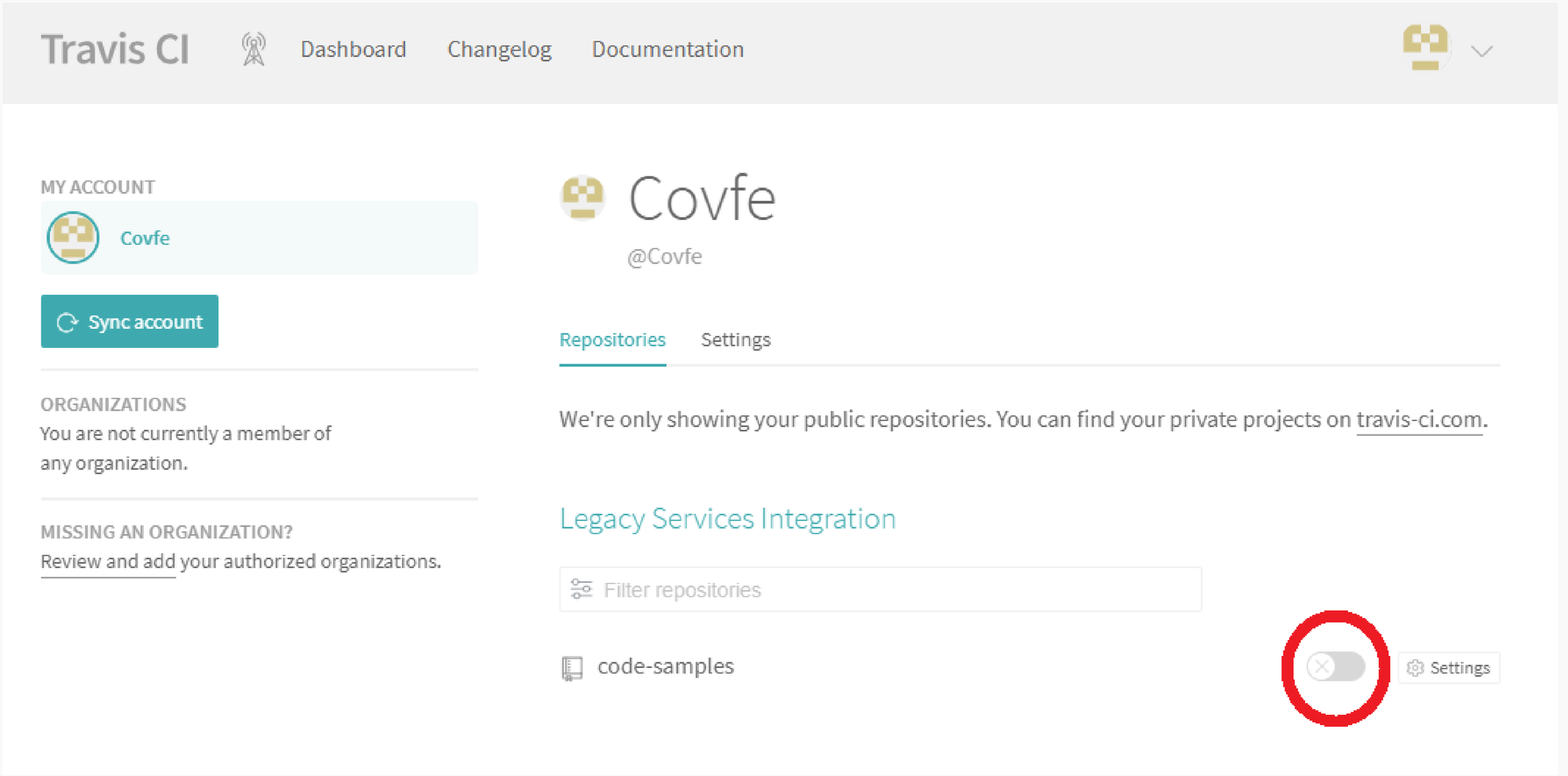
travis-ci.org/







Setting up Travis CI



Defining a .travis.yml file

In the .travis.yml a file you're specifying your **OS**, the programming **language**, your repo **branch**, the project file name and other details.

In the requirements.txt file you define requirements to be installed.

By reading these files, Travis-CI create a virtual environment with the specified parameters to run our code.

```yaml
dist: trusty
sudo: required

language: python
python:
  - "2.7.14"

install:
  - pip install --upgrade setuptools
  - pip install -r requirements.txt
  - pip install pytest pytest-cov
  - pip install coveralls

script:
  - python test_palindromes.py
  - pylint palindromes.py
  - pycodestyle palindromes.py
  - py.test --cov=.
  - coverage report

after_success:
  coveralls
```

```
pylint
pycodestyle
```

Working with CI Workshop

Follow Along!

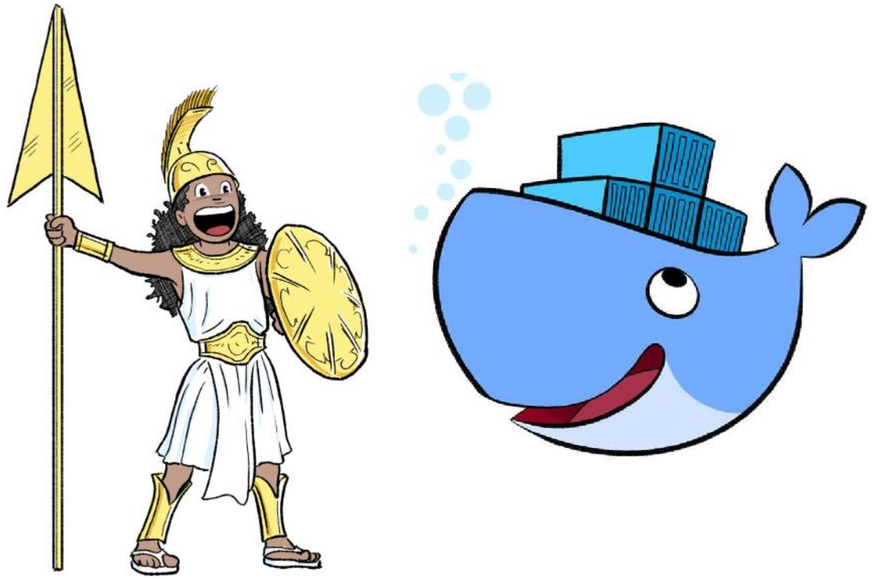
Demo time!

Beyond Continuous Integration

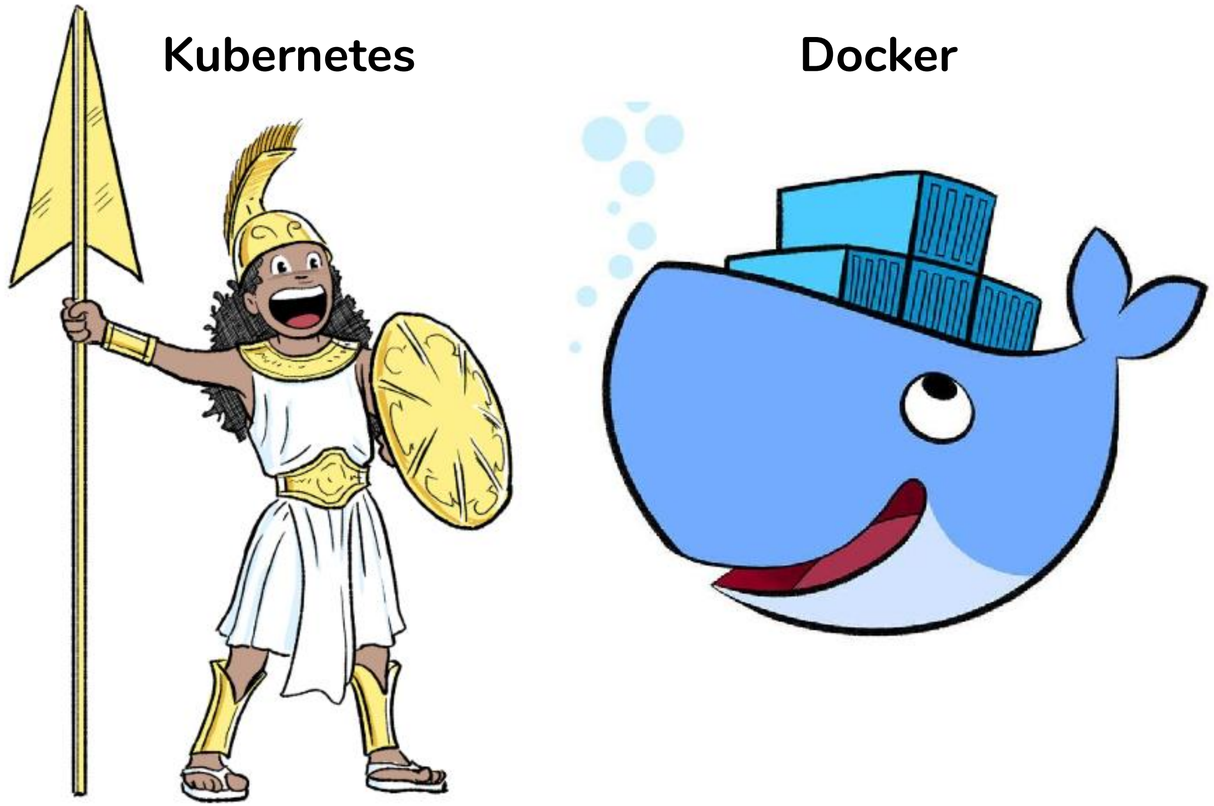
That’s cool – we can continuously test and push code into production! What’s next?

What about:
- Running your project on any platform
- Rapid deployment of applications
- Scaling your application up and down to user needs

Answer questions like:
- What are containers? What is a microservice?
- Whats a **helmsman** and a **whale** gotta do with it?



PART II OF **PRODUCTION READY CODE SERIES**



Continuous Deployment

PRESENTED BY CSEC