Devops Study Night
CICD
Stacy Watts

Senior MSS Platform Engineer
BlueVoyant

Wednesday, September 11, 2019

Wifi: VevoGuest
pw: on the whiteboard

Thank you vevo!

About Women Who Code

We are a global nonprofit dedicated to inspiring women to excel in technology careers. Our events offer study groups, technical workshops, hackathons, networking events, panel discussions, lightning talks, and social events featuring influential tech industry experts, innovators, and investors. We help you build the skills you need to raise your professional profile and achieve greater career success. Current and aspiring coders are welcome.

Code of Conduct

WWCode is an inclusive community, dedicated to providing an empowering experience for everyone who participates in or supports our community, regardless of gender, gender identity and expression, sexual orientation, ability, physical appearance, body size, race, ethnicity, age, religion, socioeconomic status, caste or creed.

...Code of Conduct

Our events are intended to inspire women to excel in technology careers, and anyone who is there for this purpose is welcome. We do not tolerate harassment of members in any form.

...Code of Conduct

Our Code of Conduct applies to all WWCode events and online communities. Read the full version at http://www.womenwhocode.com/codeofconduct. If you would like to report an incident, please reach out to one of our volunteers or submit an incident report form: http://bit.ly/wwcode-incident-report

A look at CICD Engineer Job Posts

Actual job listing bullet point

— System/Application integration and deployment using CICD tools and automation frameworks (e.g. Gitlab CI, Jenkins)

— Understanding of CICD process and tools – Jenkins, Automation, GIT

— Experience with CICD tools like Travis , Circle CI

— We work with best in class tooling like Jenkins, Circle CI, GitLab, GitHub …

These skills can be used in any devops type job, but they can also be used for development

Job titles:

— Senior Engineer - CI/CD - Java/Python/REST API

— Front-end Developer

— Back-end Developer

— Full-stack Developer

— CI/CD Engineer

— DevOps Engineer

CICD

CI Definition

Continuous Integration (CI) is a development practice.
Developers to integrate code into a shared repository several times a day.
Each check-in is then verified by an automated build.
This allows teams to detect problems early.
Integrating regularly, you can detect errors quickly.
Location detection is quicker too as the build failure is tied to a commit.

CD Definition

Continuous Delivery is getting changes into production safely, quickly and sustainably.
Changes include: new features, configuration changes, bug fixes and experiments
Developers ensure code is always in a deployable state.

Prerequisites for CI

For CI the prerequisite should be tests.
While tests are not required to set up a pipeline it isn't a good idea to set up a build pipeline without at least one legitimate test of the code.
Ideally your test suite already exists and passes on a regular basis. If it does not engineers may become quickly frustrated with the new pipeline rather than the tests themselves.
While you can develop your CI pipeline in parallel to developing robust tests make sure there are enough in place for a comfortable starting place.

Prerequisites for CD

For CD the prerequisite should be a mature build + deployment process which does not break often.
Identify all the steps it takes to deploy that code to production

Typical flow

— Version checkin

— Build

— Unit Test

— Deploy

— Automatic tests (Integration tests)

— Deploy to prod

    — possibly blue green deployment or canary deployments

— Validate and measure results

— Note that CI and CD are often spoken of in the same breath because they are part of the same logical flow.

— A commit to a repository can trigger the flow.

— Any break in the flow stops the flow toward production.

— Early state CICD pipelines often include a manual deploy to production click to ensure a gating until the process is certain to be mature.

1. put your code in version control

2. write tests

3. document deploy process

4. discuss cicd tool options

5. start building your pipeline

6. kaizen (continuous improvement)

Remember, every problem you have now will be amplified by automation.
If there are difficulties with, for example, developers committing master you may want to protect your master and enforce code reviews prior to instituting CICD!

Keywords to start your decision on which CICD tool to use:
Jenkins, Travis, TeamCity, CircleCI, Codeship, GitlabCI, Bamboo, Hudson, Puppet, Chef, VSTS, … you get the idea.

You can limit your search for starters by identifying what other teams at your company already use.
If exploring or starting from scratch there are several intro options for starting in any of them.
Search github for cicd example and the application name you want to try out. You will have a POC in no time.
Remember a POC isn't a production system, you want to go through a review with your security team, operations around the world, and other engineers to make sure there aren't any blockers to your choice.

Real world examples of pipelines on github

— https://github.com/ruby/ruby

    — ruby's travis-ci config file

— https://github.com/python/cpython

    — python's travis-ci config file

— https://github.com/CircleCI-Public/circleci-demo-go

    — CircleCI example

Complete reference docs for a few of our examples listed:

— https://docs.travis-ci.com/

— https://circleci.com/docs/2.0/configuration-reference/

— https://jenkins.io/doc/

— https://www.jetbrains.com/help/teamcity/teamcity-documentation.html

— https://confluence.atlassian.com/bamboo

— https://docs.gitlab.com/ee/ci/yaml/

Thank you!

Books

— The Phoenix Project: A Novel About IT, DevOps, and Helping Your Business Win

— The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations

— The Web Application Hacker's Handbook

— Epic Failures in Devsecops

— Accelerate: The Science of Lean Software and DevOps Building and Scaling High Performing Technology Organizations

Practice Exercise (CI)

1. sample python repo with tests

2. travis-ci demo walkthrough

3. travis docs

We will be working through 2. using 1. as our base.
Begin by forking 1. in your github account.
We will work through setting up travis from here.