Open Source and the Revolution of Software Testing

Elizabeth K. Joseph
@pleia2

Ohio LinuxFest 2018

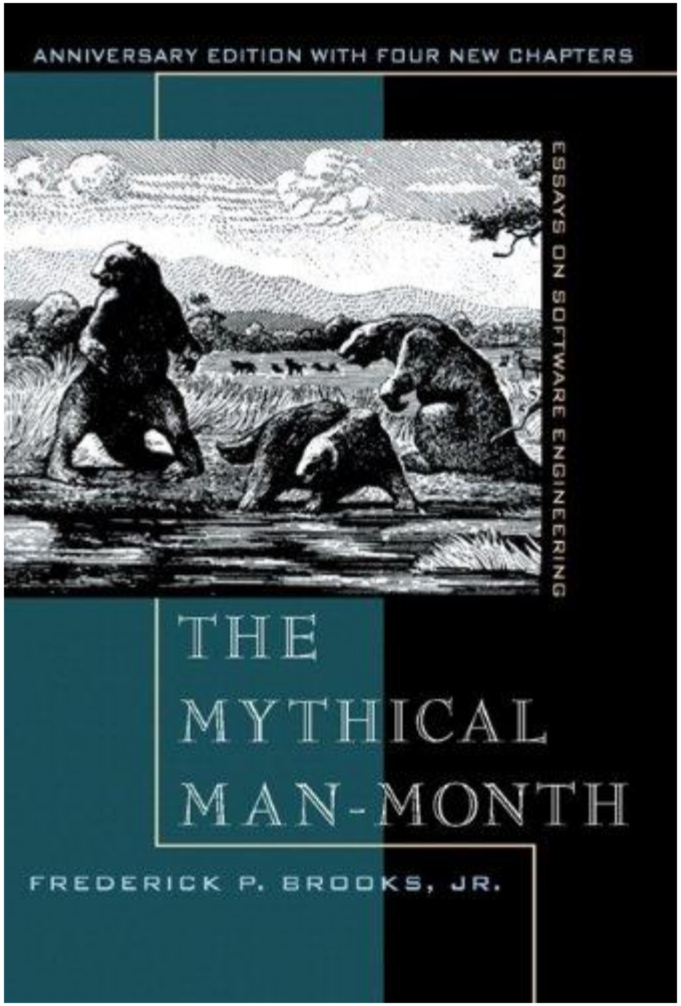
Elizabeth K. Joseph

@pleia2

- I spoke at OLF in 2016!*
- 15+ years working in and around open source communities
- 10+ years in Linux systems administration and engineering roles
- 4 years working on CI/CD for OpenStack
- Author of The Official Ubuntu Book and Common OpenStack Deployments

* The Columbus Zoo has manatees!

I was reading a book recently


ANNIVERSARY EDITION WITH FOUR NEW CHAPTERS
ESSAYS ON SOFTWARE ENGINEERING
THE MYTHICAL MAN-MONTH
FREDERICK P. BROOKS, JR.

Software Testing

“Software testing is an investigation conducted to provide stakeholders with information about the quality of the software product or service under test.”

https://en.wikipedia.org/wiki/Software_testing

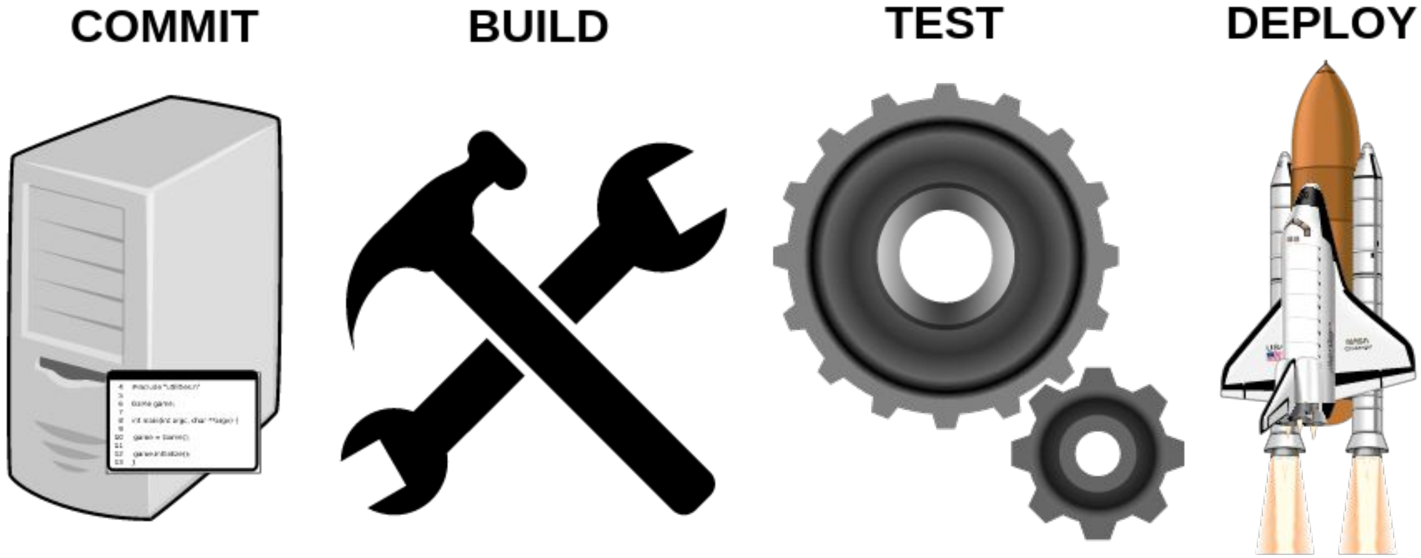
Continuous Integration (CI)

"In software engineering, continuous integration (CI) is the practice of merging all developer working copies to a shared mainline several times a day."

https://en.wikipedia.org/wiki/Continuous_integration

Common components of CI

- Keep code in a revision control system
- Automate the pipeline
- Maintain transparency across the team
- Automatically deploy (continuous deployment or delivery)



When software testing meets CI

Traditional software testing

Completed portions of software are “sent to QA”

A frequently opaque report is returned within a matter of days ...or weeks

QA largely isolated from development

With continuous integration

As small bits of code is being committed, it’s being tested

Developers get immediate feedback as to test failure/success

Developers work with QA to develop tests while the software is being written

History of open source and software testing

Open source software projects have been innovation leaders in software development.

- Distributed collaboration
- Small commits
- Descriptive commit messages

Software testing was traditionally not one of them.

History of open source and software testing

Software testing and QA as a field stretches back decades, and yet...

Open source software projects long suffered from:

- No testing at all
- Testing exclusively done inside of companies
- Maintainers “test it locally”
- Failure to integrate testing into the open source process

Turns out, we need public software testing

Clarity of process

Enforcement of coding standards

Integration testing

Security vulnerability detection

The open source community delivers!

Jenkins - https://jenkins.io/

GitLab - https://about.gitlab.com/

BuildBot - http://buildbot.net/

Spinnaker - https://www.spinnaker.io/

Artifactory - https://jfrog.com/artifactory/

Zuul - https://zuul-ci.org/

And proprietary companies do too!

Travis CI - https://travis-ci.org/

CircleCI - https://circleci.com/

TeamCity - http://www.jetbrains.com/teamcity/

GitHub integrations - https://github.com/marketplace/category/continuous-integration

...and dozens more

Continuous Integration Configuration

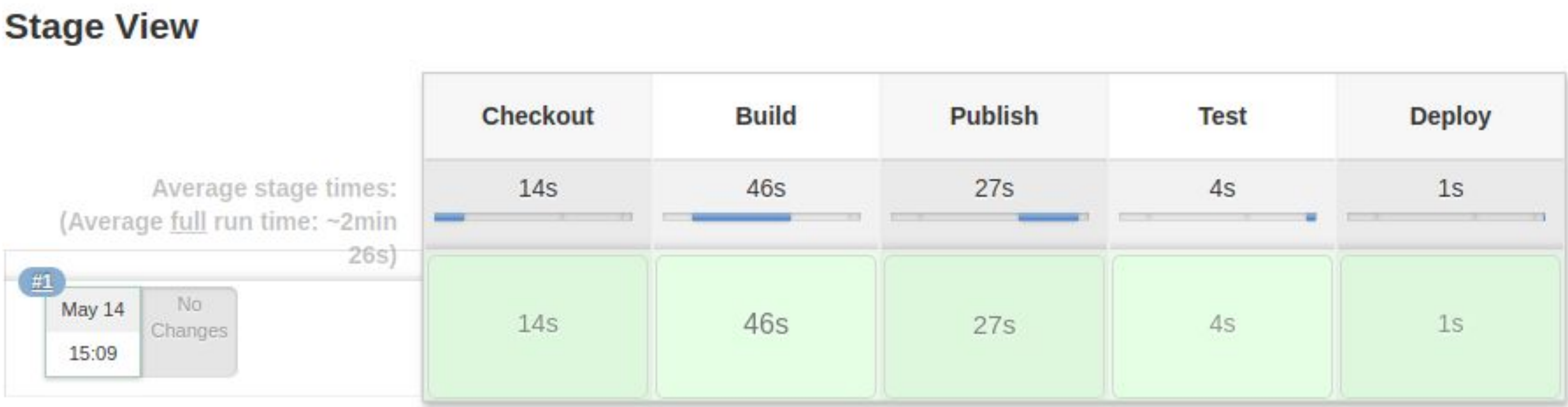
Jenkinsfile

```
node {

    // Checkout source code from Git

    stage 'Checkout'

    checkout scm

    // Build Docker image

    stage 'Build'

    sh "docker build -t cd.example.com:50000/root/site-test:${gitCommit()} ."

    // Log in and push image to GitLab

    stage 'Publish'

...
```

Circle CI

```
    steps:

      - checkout

      - run:

          command: echo 127.0.0.1 devhost | sudo tee -a /etc/hosts

      - run: |

          sudo -u root…

      # Save artifacts

      - store_artifacts:

          path: /tmp/artifacts

          destination: build

...
```

Your pipeline succeeds! (Jenkins)

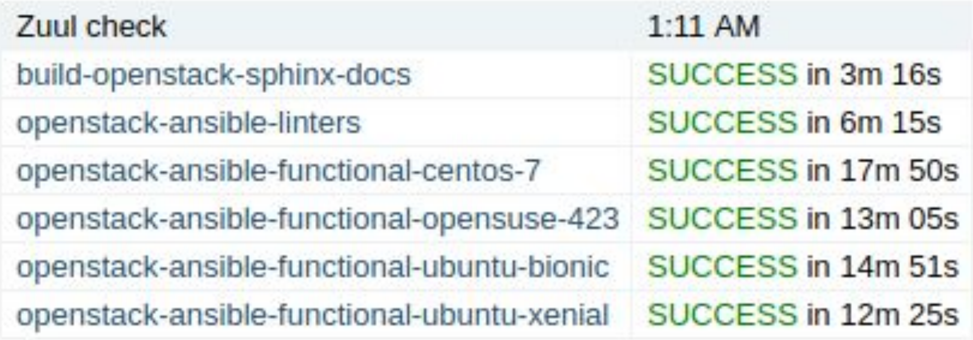


Stage View

	Checkout	Build	Publish	Test	Deploy
Average stage times: (Average full run time: ~2min 26s)	14s	46s	27s	4s	1s
#1 May 14 15:09 No Changes	14s	46s	27s	4s	1s

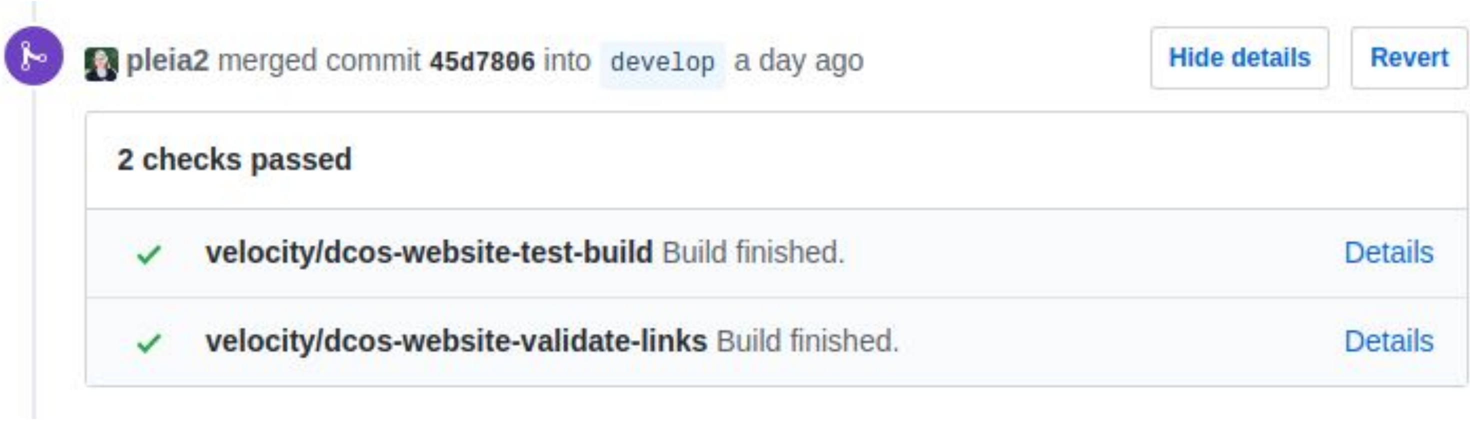
Test result reporting

Gerrit



Zuul check	1:11 AM
build-openstack-sphinx-docs	SUCCESS in 3m 16s
openstack-ansible-linters	SUCCESS in 6m 15s
openstack-ansible-functional-centos-7	SUCCESS in 17m 50s
openstack-ansible-functional-opensuse-423	SUCCESS in 13m 05s
openstack-ansible-functional-ubuntu-bionic	SUCCESS in 14m 51s
openstack-ansible-functional-ubuntu-xenial	SUCCESS in 12m 25s

GitHub



pleia2 merged commit 45d7806 into develop a day ago

2 checks passed

✓ **velocity/dcos-website-test-build** Build finished. Details

✓ **velocity/dcos-website-validate-links** Build finished. Details

Add testing to YOUR project

Every open source project maintainer I spoke with wished they could have automated testing, but lacked one or all of the following:

- Time
- Expertise
- Other resources (including infrastructure and money)

So, where to begin?

The proprietary route: Benefits

- The value add of services is making it easier
- Easier to configure and integrate
- Includes infrastructure-side maintenance
- May abstract out job running
- Initial monetary costs are low

The proprietary route: Considerations

- Is the service I’m using adhering to open standards, or am I locked in?
- Does the vendor have a history of communicating clearly and honestly with their customers?
- How flexible is their infrastructure, and will you need to make compromises to fit into their tooling?
- Does the vendor respond to bugs and feature requests?
- Will the vendor use our data in a way that I’m not comfortable with?
- Initial costs may be zero, or low, but do you have a plan to handle long term, growing costs?

The open source route: Benefits

- Open source! And everything that comes with that.
- Full control of your infrastructure and data
- Flexibility in what you want to run
- Build up expertise in a booming field

The open source route: The path

- Put out a call for systems administrators in your community who can help. They exist. I was one of them!
- Familiarize yourself with the open source CI tooling ecosystem
- Look at what other projects are doing
  - https://opensourceinfra.org
  - https://openci.io/
- Leverage free resources available to open source projects. Many cloud providers have them.
- Partner with a larger project that already has the resources, or a non-profit in your space.

Welcome to the revolution.

Happy testing!

Contact

Elizabeth K. Joseph

lyz@princessleia.com

@pleia2

Open source infrastructure efforts:

- https://opensourceinfra.org/
- https://openci.io/