Continuous Delivery

Michał Knapik

u2i

Continuous Delivery

Agenda

- rationale
- foundations & principles
- pipeline
- benefits

Software Engineering

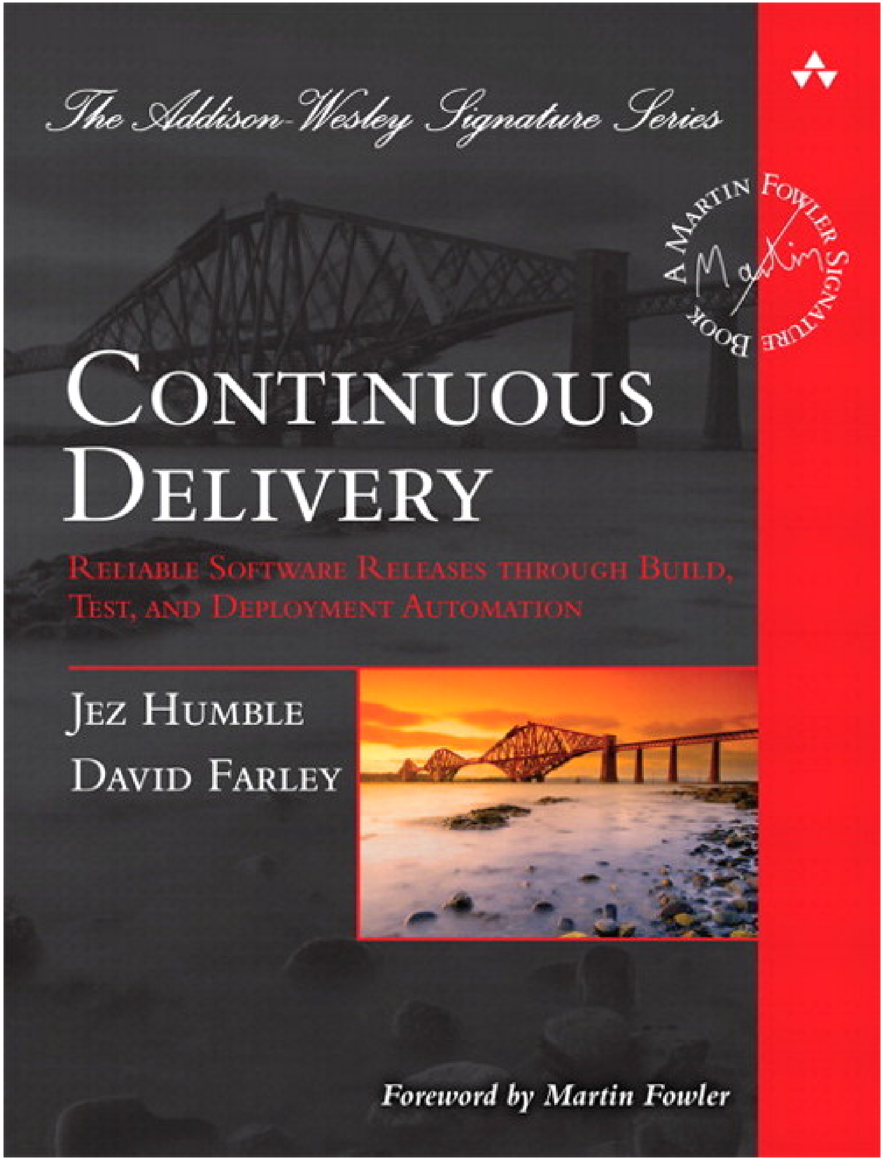
Software Engineering

> “ If bridge building were like programming, halfway through we’d find out that the far bank was now 50 meters farther out, that it was actually mud rather than granite, and that rather than building a footbridge we were instead building a road bridge.
>
> *— Sam Newman, Building Microservices*

Agile Manifesto

Agile Manifesto

> (…) Our highest priority is to satisfy the customer through early and **continuous delivery** of valuable software. (…)
>
> — *Agile Manifesto, 2001*



The Addison-Wesley Signature Series

A Martin Fowler Signature Book

Martin

Continuous Delivery

Reliable Software Releases through Build, Test, and Deployment Automation

Jez Humble

David Farley

Foreword by Martin Fowler

Continuous Delivery

> “ Continuous Delivery is the ability to get changes of all types—including new features, configuration changes, bug fixes and experiments—into production, or into the hands of users, safely and quickly in a sustainable way.
>
> *— continuousdelivery.com*

Continuous Delivery

Ingredients

- version control
- continuous integration
- automation
- configuration management
- responsibility

Version Control

Keep everything in version control!

- code
- configuration
- infrastructure setup

Version Control

Continuous Integration

Continuous Integration

It's a practice, not a tool!

Version Control

Continuous Integration

Continuous Integration

It's a practice, not a tool!

- Merge to master/trunk as often as possible
- Avoid long-living branches
- Use feature toggles if necessary

Version Control

Continuous Integration

Automation

development

Development Setup

Version Control

Continuous Integration

Automation

development

Development Setup

ANSIBLE

HashiCorp Vagrant

Version Control

Continuous Integration

Automation

development

tests

Tests

Version Control

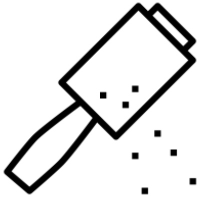
Continuous Integration

Automation

development

tests

lint



Lint

Version Control

Continuous Integration

Automation

development

tests

lint

Lint

RuboCop

BRAKEMAN

Version Control

Continuous Integration

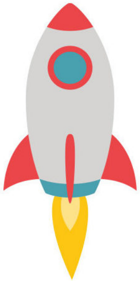
Automation

development

tests

lint

deployment



Deployment

Version Control

Continuous Integration

Automation

development

tests

lint

deployment

infrastructure

Infrastructure

Version Control

Continuous Integration

Automation

development

tests

lint

deployment

infrastructure

Infrastructure

$_
HashiCorp
Terraform
CHEF
HashiCorp
Packer

Version Control

Continuous Integration

Automation

Configuration Management

Configuration Management

Version Control

Continuous Integration

Automation

Configuration Management

Configuration Management

CHEF

HashiCorp
Consul

Version Control

Continuous Integration

Automation

Configuration Management

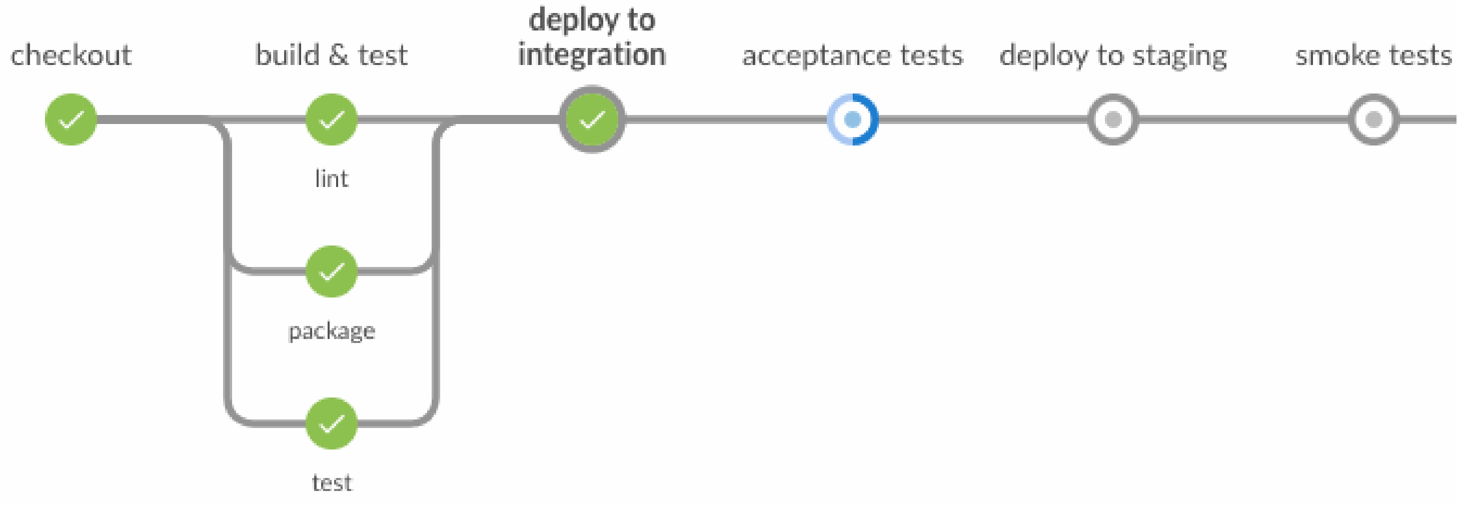
Responsibility

Responsibility

Discipline

> (…) Continuous delivery is not magic. It’s about continuous, daily improvement—the constant discipline of pursuing higher performance by following the heuristic “if it hurts, do it more often, and bring the pain forward.” (…)
>
> — *continuousdelivery.com*

Pipeline


checkout
build & test
lint
package
test
deploy to integration
acceptance tests
deploy to staging
smoke tests

Test

- write tests
- write fast tests
- avoid flaky-tests
- production-like environment

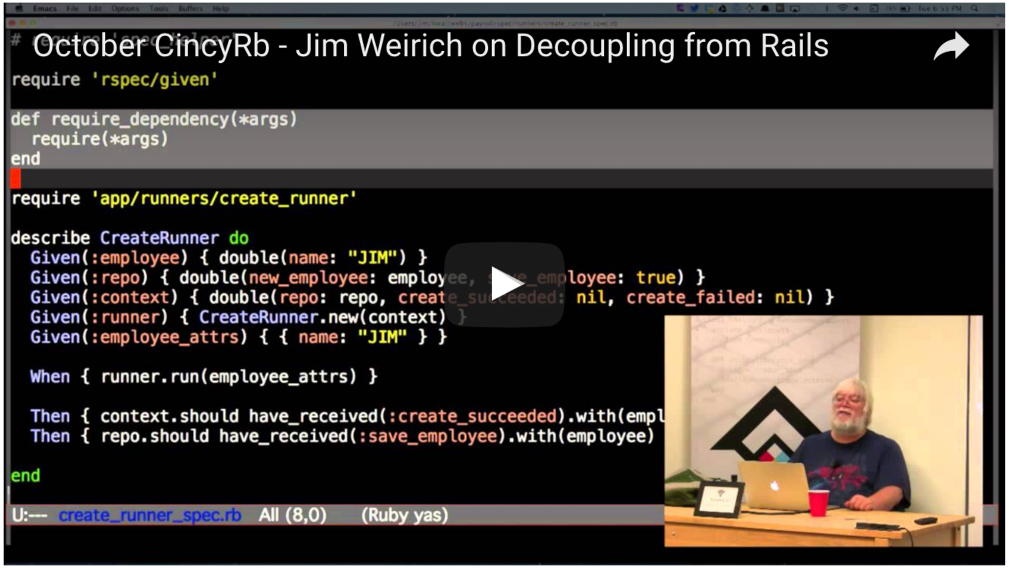
Speed

- decouple business logic
- identify slow tests
- tweak database setup
- hardware

Test

speed

Decouple Business Logic



October CincyRb - Jim Weirich on Decoupling from Rails

```
# require 'spec_helper'
require 'rspec/given'

def require_dependency(*args)
  require(*args)
end

require 'app/runners/create_runner'

describe CreateRunner do
  Given(:employee) { double(name: "JIM") }
  Given(:repo) { double(new_employee: employee, save_employee: true) }
  Given(:context) { double(repo: repo, create_succeeded: nil, create_failed: nil) }
  Given(:runner) { CreateRunner.new(context) }
  Given(:employee_attrs) { { name: "JIM" } }

  When { runner.run(employee_attrs) }

  Then { context.should have_received(:create_succeeded).with(empl
  Then { repo.should have_received(:save_employee).with(employee)

end
```

Speed

- `spec_helper.rb` vs `rails_helper.rb`
- `Bundler.require(:default, Rails.env)`
- `database_rewinder` over `use_transactional_fixtures`
- `activerecord-import`
- `rspec --profile`

Test

speed

Use Hardware

- `parallel_tests`
- `knapsack`
- CI workers autoscaling

Test

speed

flakiness

> Unfortunately, across our entire corpus of tests, we see a continual rate of about 1.5% of all test runs reporting a "flaky" result.
> ...
> Almost 16% of our tests have some level of flakiness associated with them!
> ...
> What we find in practice is that about 84% of the transitions we observe from pass to fail involve a flaky test!
>
> *— John Mico, Google*

Flaky Tests at Google and How We Mitigate Them by John Mico

Test

speed

flakiness

Fix Indeterminism

- `eq` vs `match_array`
- `Time.parse`, `Time.now`
- Rails autoload and test order Rails Autoloading Hell by Simon Coffey, Urban Automaton
- random is not unique – `SecureRandom.uuid`
- use `Tempfile`

Test

speed

flakiness

Fix Indeterminism

RSpec

- `config.order = :random`
- `rspec --bisect`
- avoid `before(:all)`

Test

Package

Artifact

Test

Package

Artifact Repository

GEMSTASH
archiva
Sonatype
Nexus

Test

Package

Formats

- git ref
- gem
- container
- system package

Test

Package

gem

Gem

Use a package manager native to an environment.

Test

Package

gem

Gem

Use a package manager native to an environment.

Test

Package

gem

container

Test

Package

gem

container

Containers

Test

Package

gem

container

system packages

System Packages

Test

Package

gem

container

system packages

System Packages

- fpm
- pkgr

Test

Package

gem

container

system packages

System Packages

- fpm
- pkgr

Test

Package

Deploy

Deploy

Test

Package

Deploy

Deploy

> Note: This app is intended for people with experience deploying and maintaining Rails applications. If you're uncomfortable with any steps below then X is not for you.
>
> *— Open Source Rails Application, Github*

Runtime Environment

Use system packages whenever possible.

Replace RVM with deb package:
https://www.brightbox.com/docs/ruby/ubuntu/

Test

Package

Deploy

runtime environment

artifact repository

Utilize The Artifact Repository

- `gem install the-app`
- `docker run the-app`
- `apt install the-app`

Test

Package

Deploy

runtime environment

artifact repository

health check

Health Check

```
{
  "version": "1.1.2-1024",
  "healthy": false,
  "db": true,
  "cloud-credentials": false
}
```

A simple `curl` might be good enough

Acceptance Tests

- API
- UI

Test

Package

Deploy

Acceptance Tests

Acceptance Tests

- API
- UI

cucumber

Se

Test

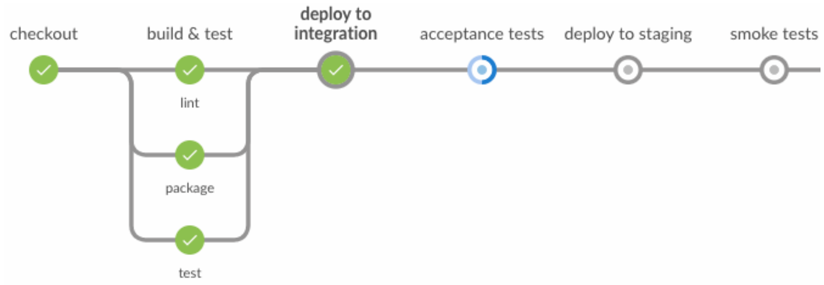
Package

Deploy

Acceptance Tests

Promotion

Promotion



Test

Package

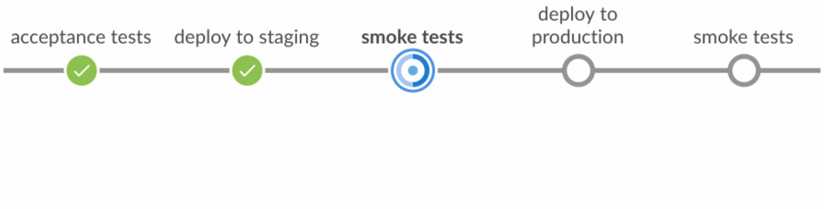
Deploy

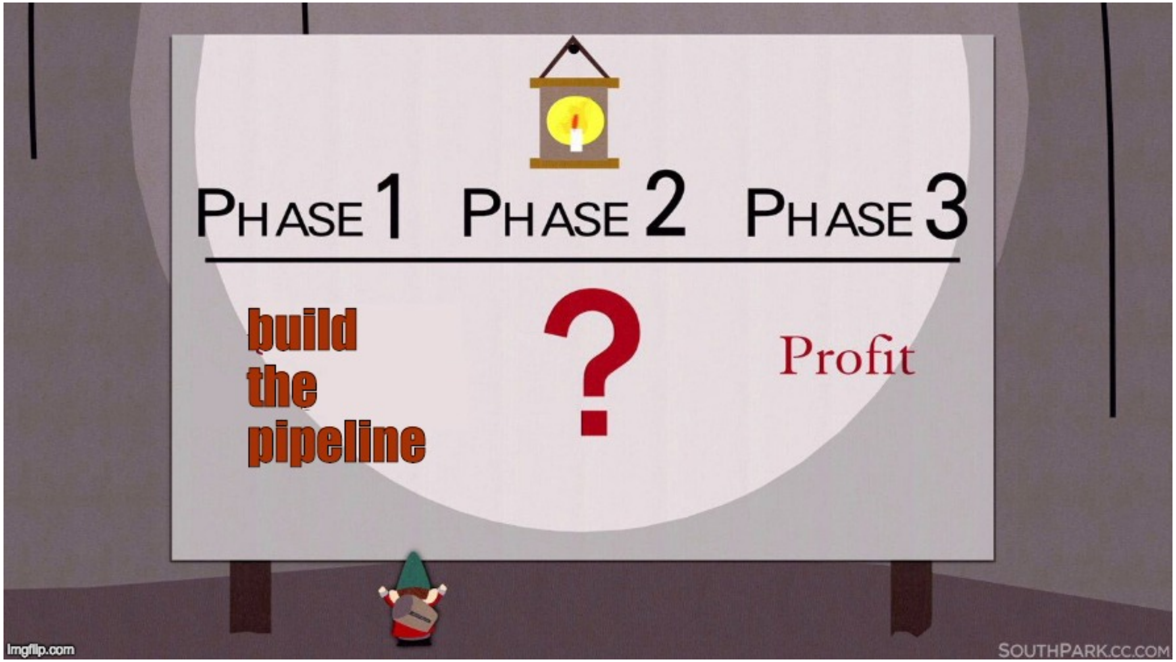
Acceptance Tests

Promotion

Smoke Tests

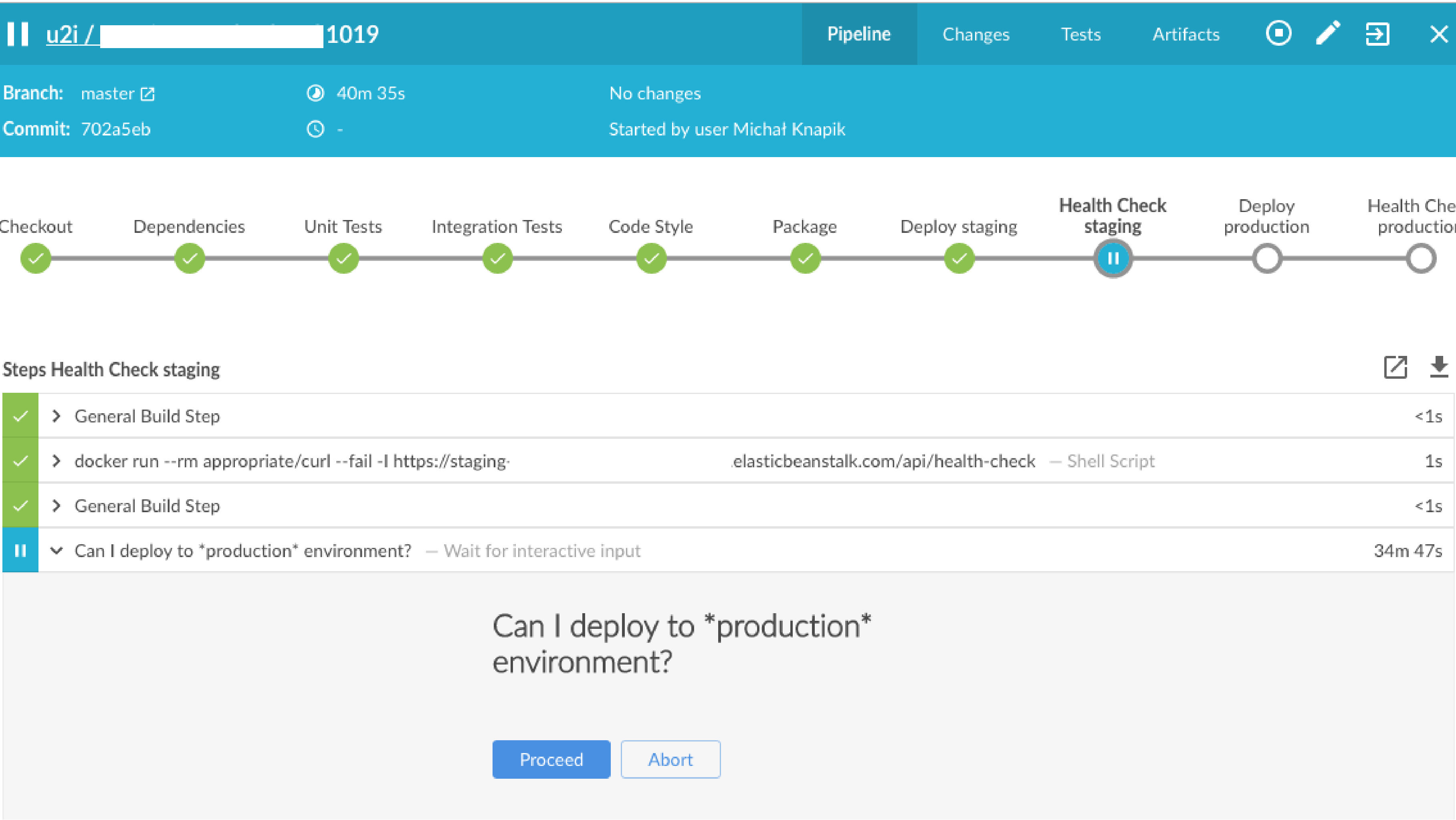
Smoke Tests




PHASE 1
PHASE 2
PHASE 3
build the pipeline
?
Profit
imgflip.com
SOUTHPARK.CC.COM

Benefits

- Mean Time To Recovery
- velocity
- confidence & quality
- stress



u2i / [redacted]1019

Pipeline | Changes | Tests | Artifacts

Branch: master

Commit: 702a5eb

40m 35s

-

No changes

Started by user Michał Knapik

Checkout ✓ — Dependencies ✓ — Unit Tests ✓ — Integration Tests ✓ — Code Style ✓ — Package ✓ — Deploy staging ✓ — **Health Check staging** — Deploy production ○ — Health Che production ○

Steps Health Check staging

✓	General Build Step	<1s
✓	docker run --rm appropriate/curl --fail -I https://staging-[redacted].elasticbeanstalk.com/api/health-check — Shell Script	1s
✓	General Build Step	<1s
‖	Can I deploy to *production* environment? — Wait for interactive input	34m 47s

Can I deploy to *production* environment?

References

- https://continuousdelivery.com
- https://testing.googleblog.com/2016/05/flaky-tests-at-google-and-how-we.html
- https://commons.wikimedia.org/wiki/File:Git_icon.svg
- https://commons.wikimedia.org/wiki/File:Team_work_icon.svg
- https://www.cmcrossroads.com/article/why-you-need-be-doing-continuous-integration
- https://rspec.info/
- https://www.deviantart.com/superawesomevectors/art/Rocket-Ship-Flat-Vector-Icon-696350854
- http://icons-for-free.com/icon/computers_servers_web_host_icon_286686.html
- https://en.wikipedia.org/wiki/File:Chef_logo.svg#file
- https://github.com/odb/official-bash-logo
- https://www.hashicorp.com/brand
- https://commons.wikimedia.org/wiki/File:Ansible_logo.svg
- https://12factor.net/
- https://www.onlinewebfonts.com/icon/463475
- https://github.com/ArturT/knapsack
- https://rubygems.org/
- https://www.docker.com/legal/brand-guidelines
- https://www.atlassian.com/blog/archives/docker-all-the-things-at-atlassian-automation-and-wiring
- https://github.com/bundler/gemstash
- https://www.jfrog.com/confluence/display/RTF/Welcome+to+Artifactory
- https://github.com/cncf/landscape/issues/157
- https://cucumber.io
- https://testingrepository.com/capybara-increase-max-allowed-page-load-time/
- https://github.com/rubocop-hq/rubocop
- https://github.com/prontolabs
- https://github.com/prettier/prettier-logo
- http://brakemanscanner.org/
- https://thenounproject.com/term/lint-roller/989975/

Thanks!

blog.knapik.me
@mknapik
@michal_knapik