Docker-native Automated Delivery

...

Stefan Thorpe, CTO

caylent™

Overview

1. Bio
2. Our Aim At Caylent
3. Why Continuous Delivery
4. CD vs CD
5. How Docker Improves CD
6. Good CD Guide
7. Deployment Types
8. Demo
9. Caylent Under The Hood

Bio

Stefan Thorpe

Co-Founder/CTO
Caylent

Linux
AWS
Bash Scripting
Python and PHP

amazon web services | Certified
Solutions Architect - Professional

RBS
The Royal Bank of Scotland Group

CISCO
CCNA

WARNER MUSIC

Our Aim

… is to support companies in delivering software more rapidly by helping them embrace DevOps and its best practices.

We believe that key to DevOps is the complete automation of Continuous Delivery and Continuous Deployment Pipelines

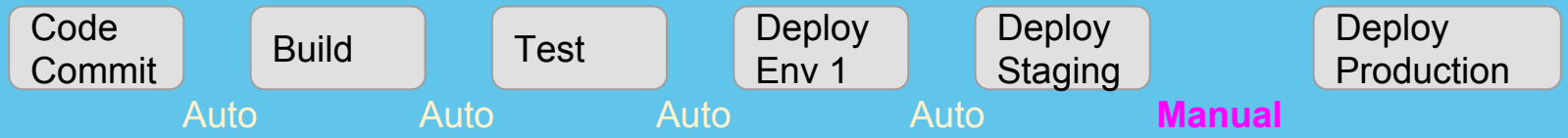
	Skilled DevOps Teams	**Non-DevOps Teams**
Deployment Frequency	On Demand (many times per day) - 200x	Between once per month and once every 6 months
Lead Times (Code Commit - Running in Production)	Less than an hour - 2555x	Between one and 6 months
Mean Time To Recovery	Less than an hour	Less than one day
Change Failure Rate	0 - 15%	31 - 45%

Why Continuous Delivery is Key to DevOps

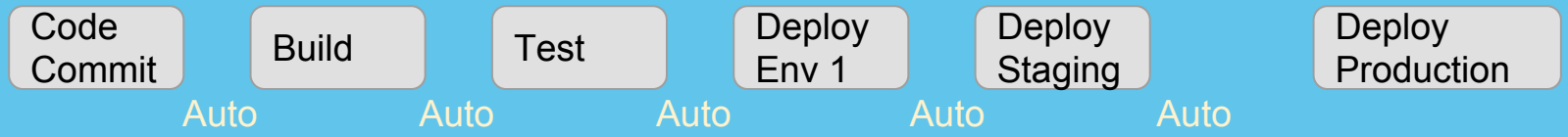
- Low risk releases
- Shorter lead times
- Higher quality
- Lower costs
- Better products
- Happier teams

Continuous Delivery vs Continuous Deployment

Continuous Delivery

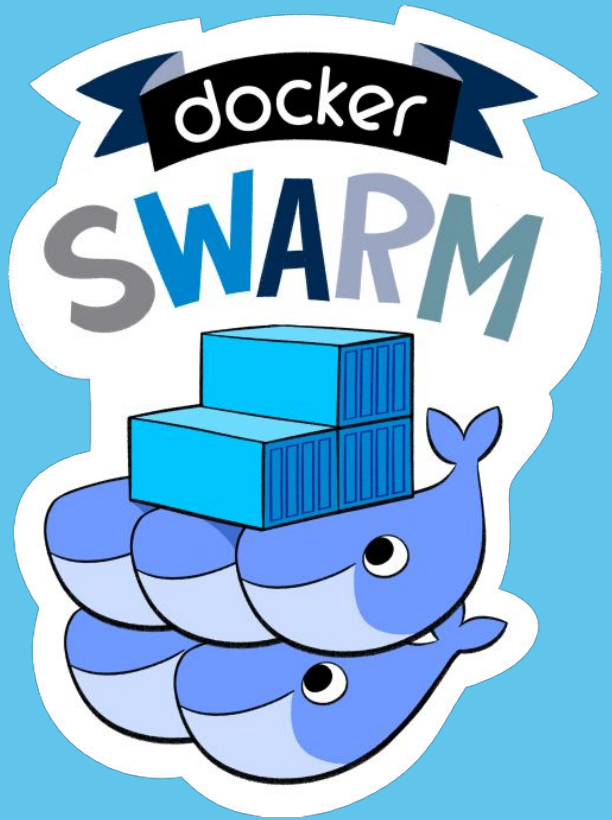


Continuous Deployment



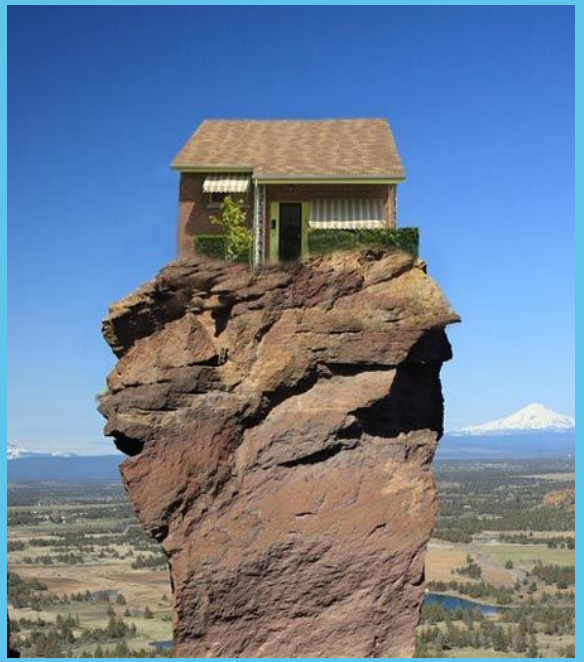
How Docker Improves CD

- Build Once - Immutable Artifacts
- Increased Robustness
- Build Once - Test Many
- Quick Roll Backs
- Simplified Deployments
- Faster Deployments
- Server Config as Code



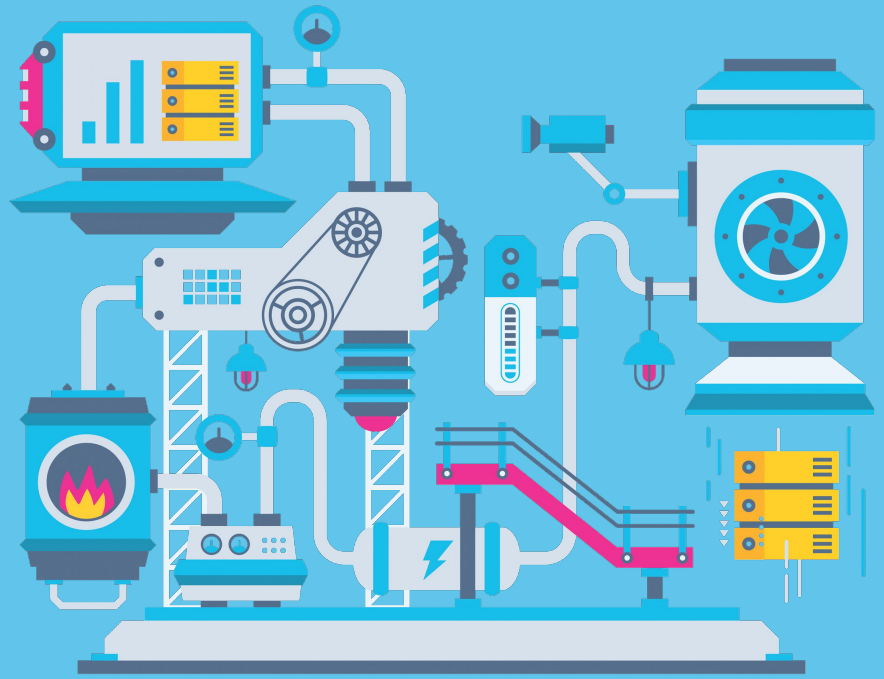
Good CD Guide

- Only build packages once
  - Leverage Docker's tagging system to promote your image
- Deploy code, database migrations and infrastructure changes separately
- Deploy the same way to each environment
  - Pass environment variables at run time - consider changes to these as Infrastructure level
- Automate testing your deployments
  - Where possible combine builds and tests together
- Keep your environments similar



Deployment Types

- Minimum in service deployment
- Rolling Application Updates
- Blue/Green
- A/B



Minimum In-Service Deployment

Deployments happens in stages

Few moving parts - Orchestration and health checks are required (Outside of Swarm)

No additional Infrastructure cost

Generally no down time

Rolling Deployments - Docker Swarm

Deployment happens in multiple stages, targets per stage user-defined

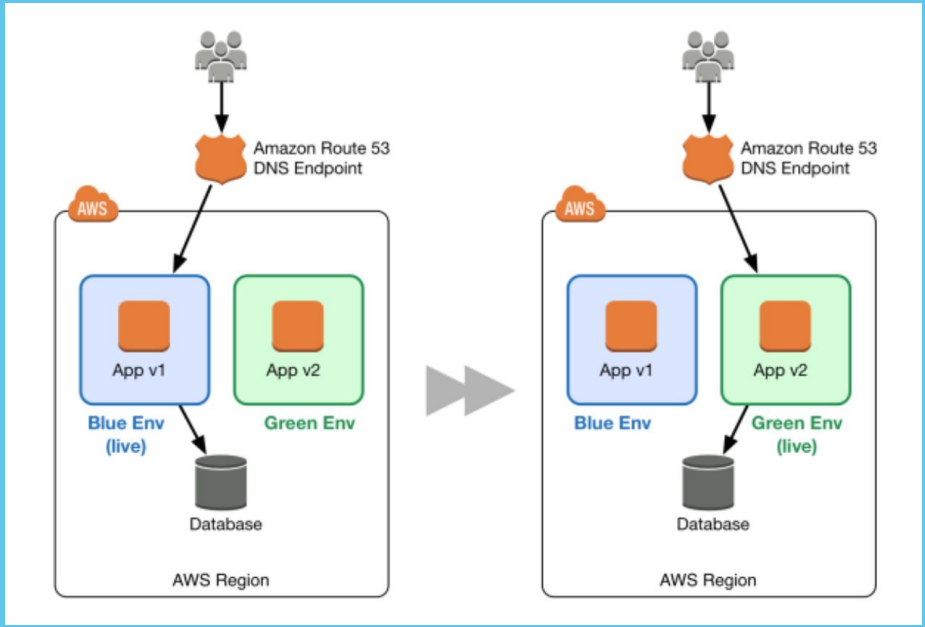
Supported by Docker Swarm (Pause and Continue only)

Orchestration and health checks are recommended (Outside of Swarm)

Can be least efficient deployment time, based on time-taken

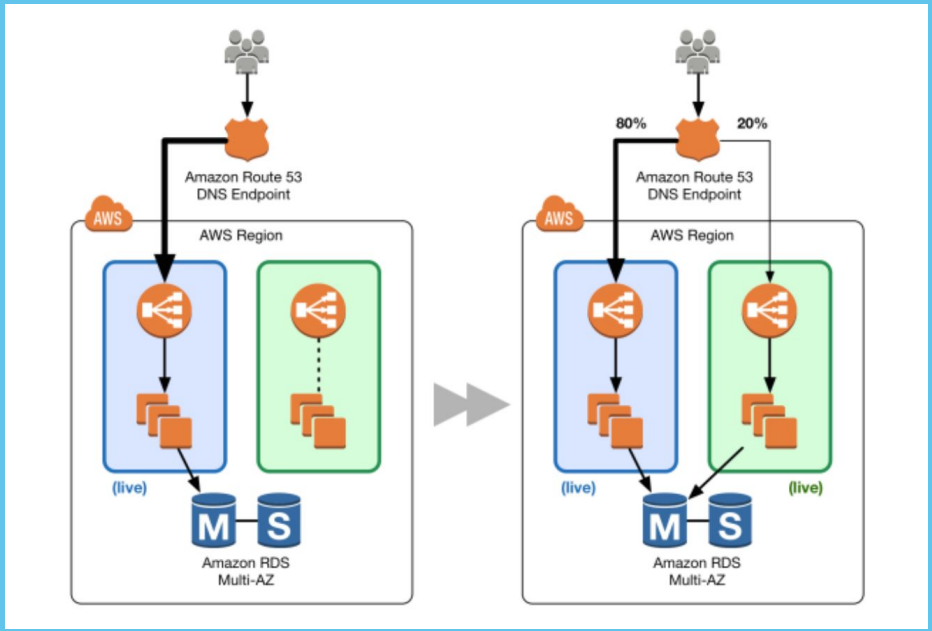
Generally no downtime

Blue/Green



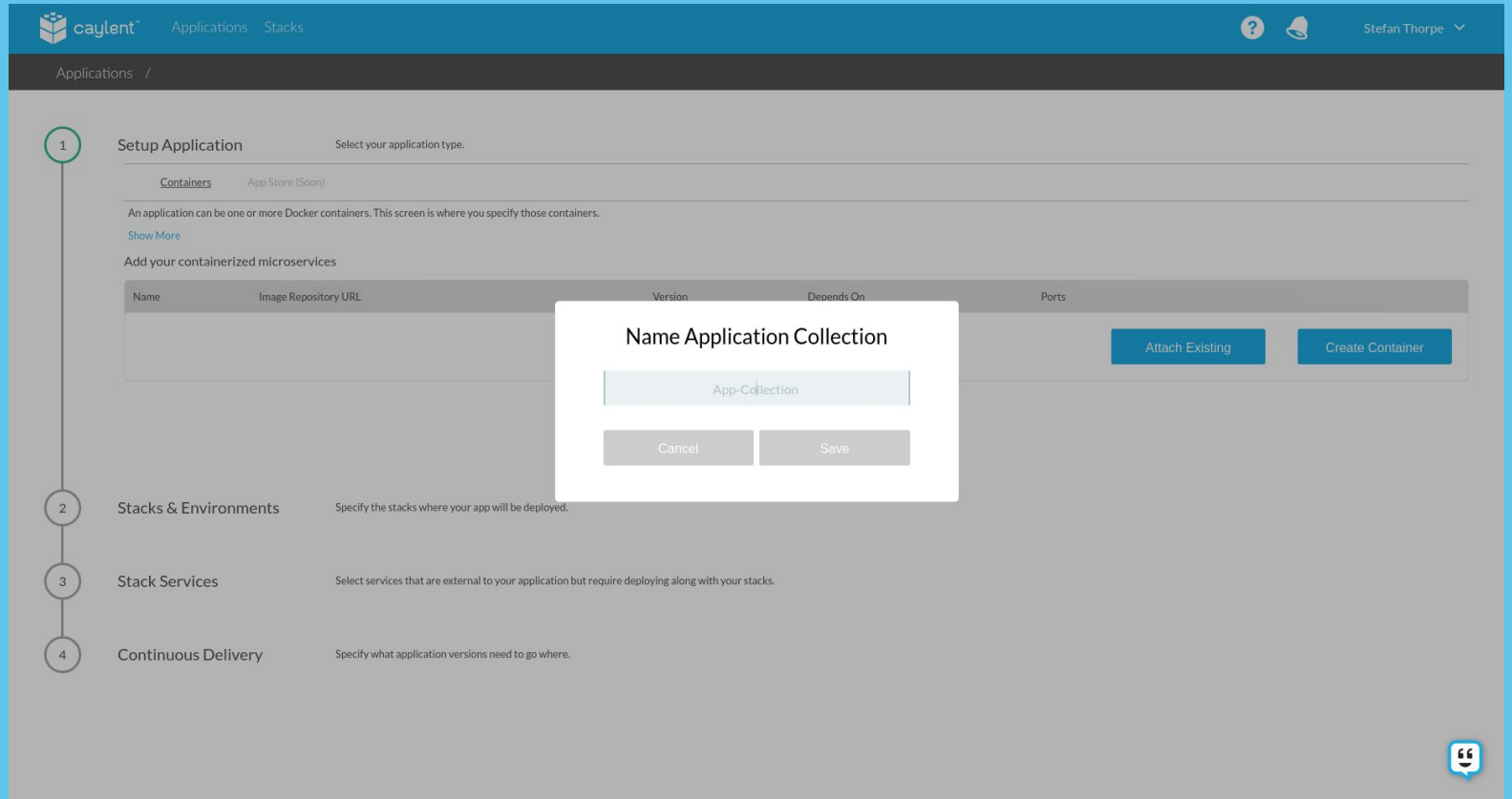
- Near zero-downtime release and rollback - DNS
- Reduced Deployment Risk
  - Isolation between blue and green environments
- Larger validation window
  - Complete health and performance tests possible
- Increased Infrastructure Cost
- ‘Zero to Hero’ Problem

A/B Deployment

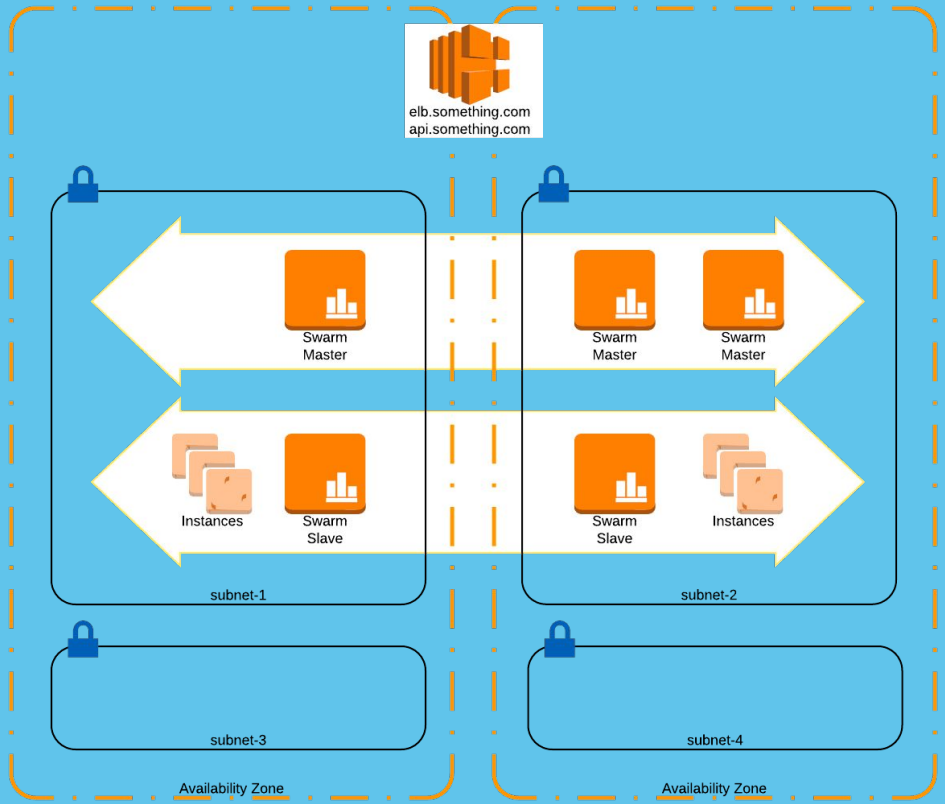


- All benefits of Blue/Green
- Customer validation
- Limited blast impact
- Capacity scaling and pre-warming
- Lots of moving parts

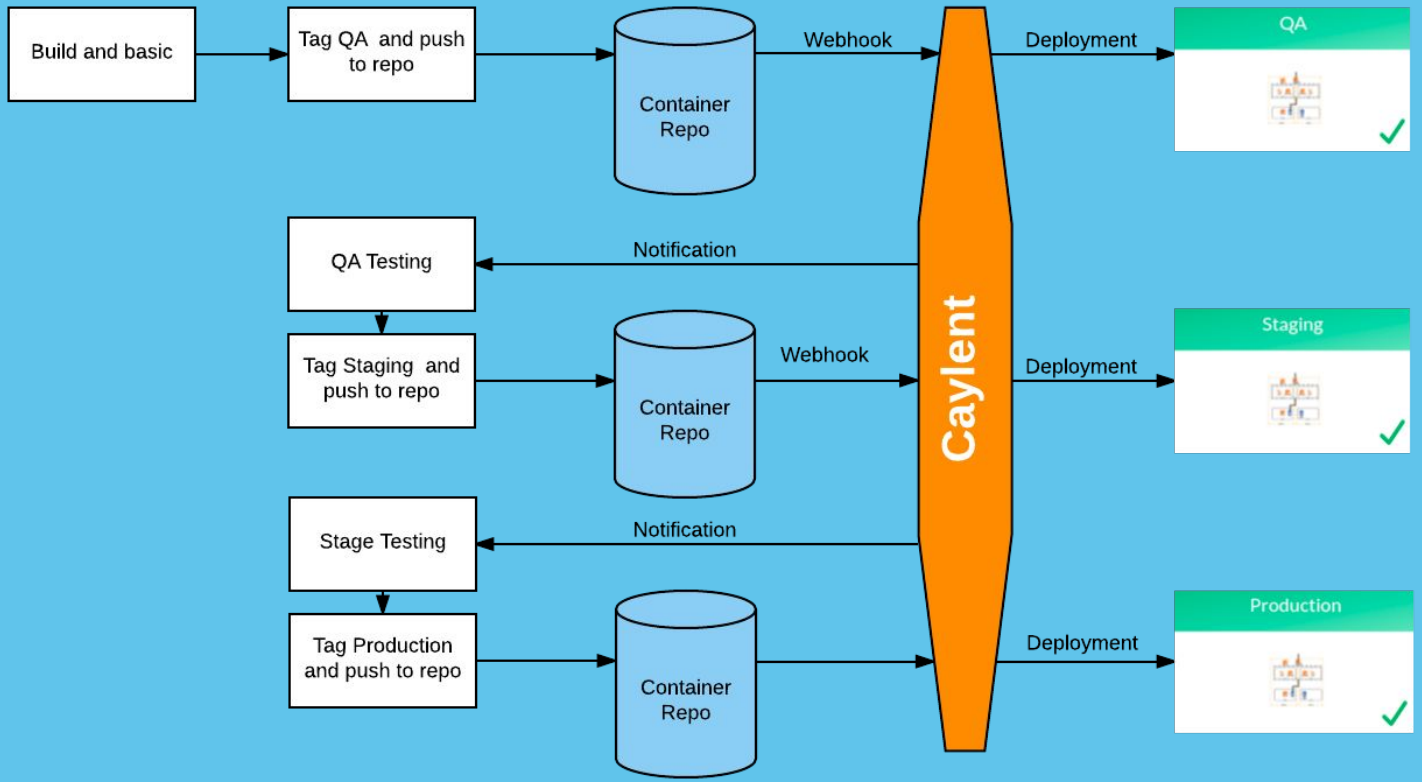
Caylent Setup / Wizard Demo



Under The Hood - AWS



Docker Hub & Caylent



Q&A

stefan@caylent.com