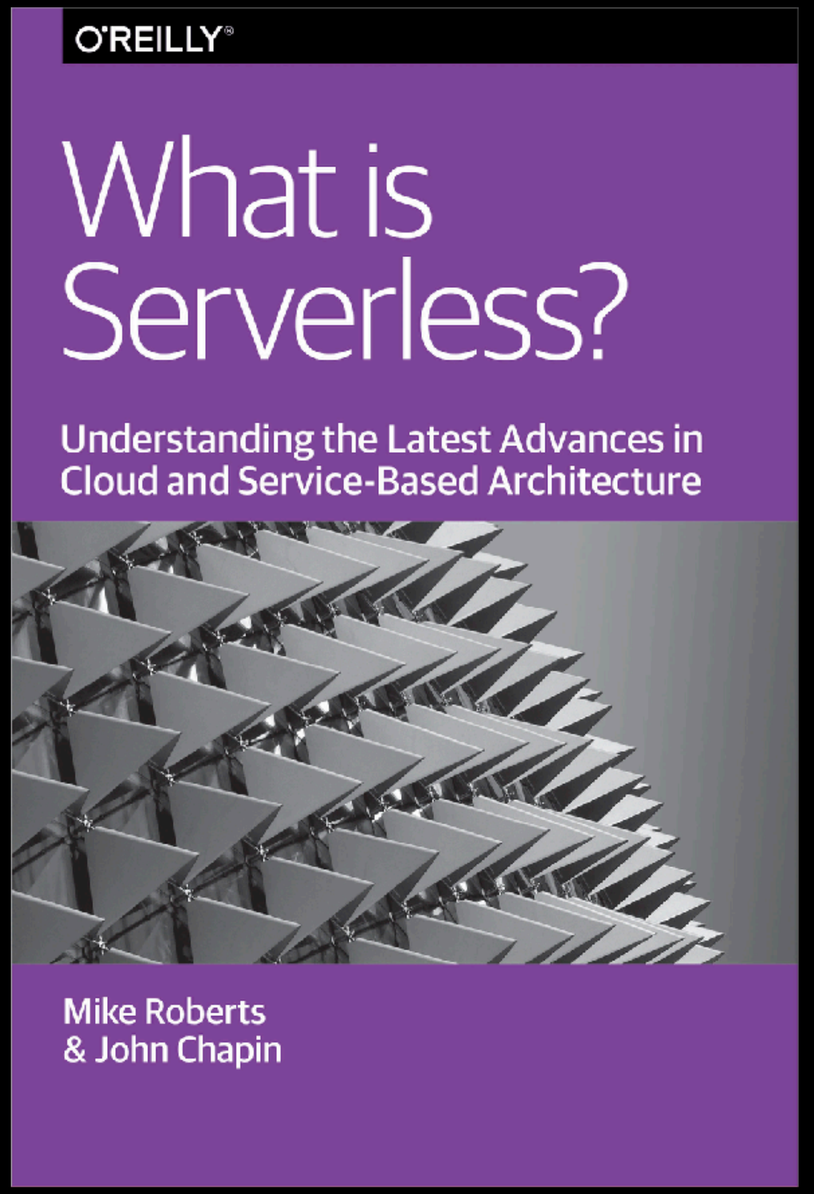
Continuous Delivery in an Ephemeral World

@johnchapin | symphonia.io

sacon@lists.symphonia.io

john@symphonia.io

sacon@lists.symphonia.io



Agenda

- Continuous Delivery refresher
- Continuous Delivery on AWS
- The Challenges of Ephemeral Continuous Delivery
- Tutorial
- Discussion and Questions

AUDIENCE PARTICIPATION

How long does it take for a commit to be deployed to production?

Less than a month?

Less than a week?

Less than a day?

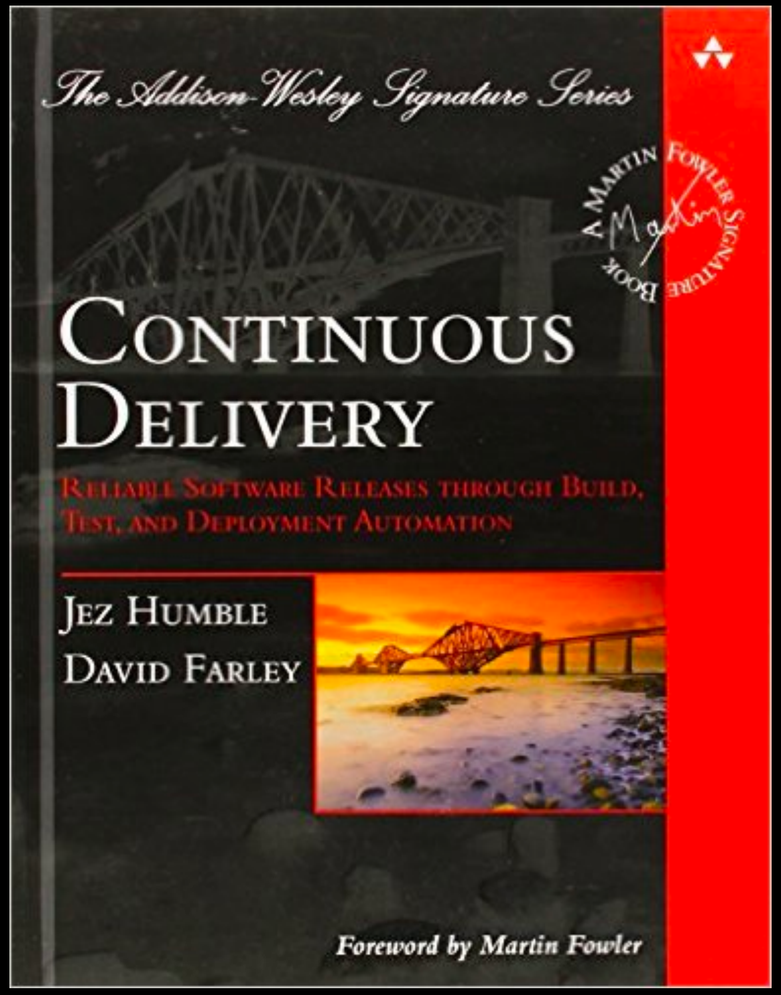
Less than an hour?

Less than a minute?

Continuous Delivery

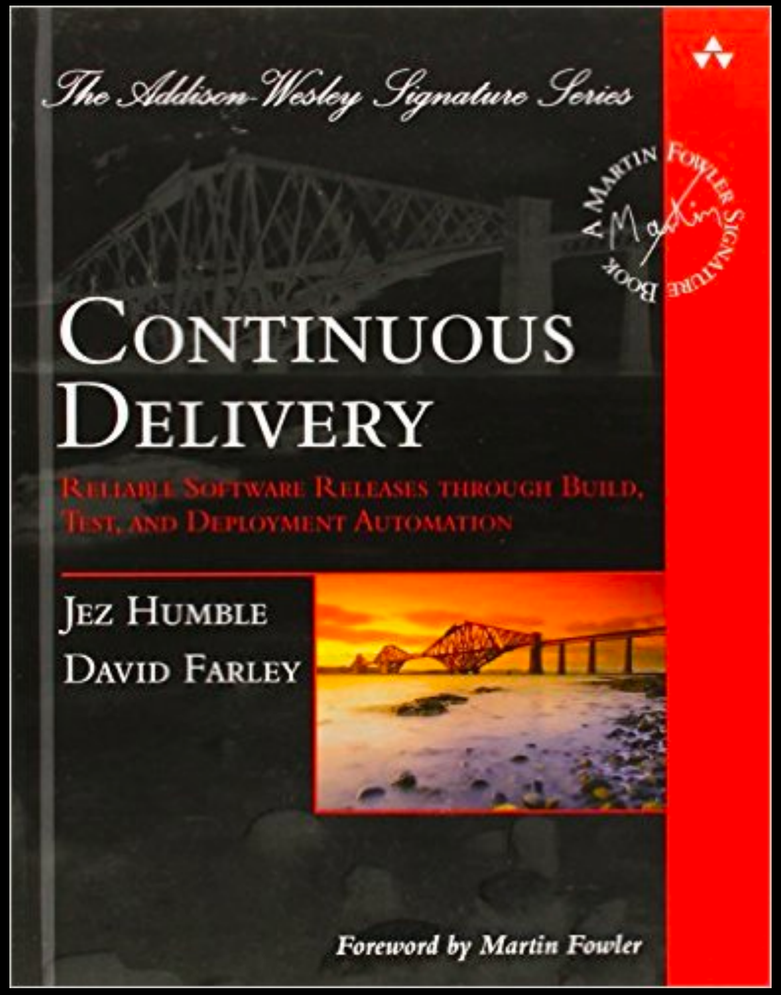
In a nutshell...

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.



Benefits

- Lower risk
- Faster time to market
- Higher quality
- Lower costs
- Better products
- Happier teams



Deployment Pipelines

- Automation of deployment from source control to any environment
- Early tools
  - Hudson
  - CruiseControl, CruiseControl.NET

Continuous Integration/Delivery Tools

- **On-premise:**
  - Jenkins
  - TeamCity
  - Bamboo
- **Hosted:**
  - TravisCI
  - CircleCI
  - Semaphore

Friends of Continuous Delivery

- Modular, decoupled systems
- Immutable infrastructure
- Infrastructure-as-code
- Monitoring

Continuous Delivery on AWS

AWS CodePipeline

- Continuous integration and delivery as a service
- Integrations with other AWS services (like CodeBuild)
- Custom actions via Lambda
- Declarative JSON (or YAML) templates

AWS CodeBuild

- Software builds as a service
- First-class support for building Java, Python, Node.js, Ruby, Go, Android, and Docker projects
- Fully custom build environments via ECR / Docker images
- Declarative YAML specifications

Friends of Continuous Delivery on AWS

- Serverless!
- CloudFormation (infrastructure-as-code)
- CloudWatch (monitoring)

Benefits

- Scalable
- API-driven
- CloudFormation-enabled
- IAM security
- Pay-as-you-go
  - CodePipeline billed per-pipeline
  - CodeBuild billed by time

Rough edges

- Web console
- Missing integrations/features
  - CodePipeline -> BitBucket
  - Slack
- CodePipeline "source" != CodeBuild "source"
- Events aren't usefully enriched

Ephemeral Continuous Delivery

System Events

1. Git commit pushed to source repository
2. Via polling or an event, pipeline is started
3. Pipeline receives or downloads source
4. Pipeline spins up containers for build/test/deploy
5. Build/test/deploy runs, passing input/output as needed
6. Pipeline tears down containers

Challenges

1. All builds start from scratch
2. Containers take time to instantiate and spin up
3. Pipeline components don't share state
4. Any state created during build is lost

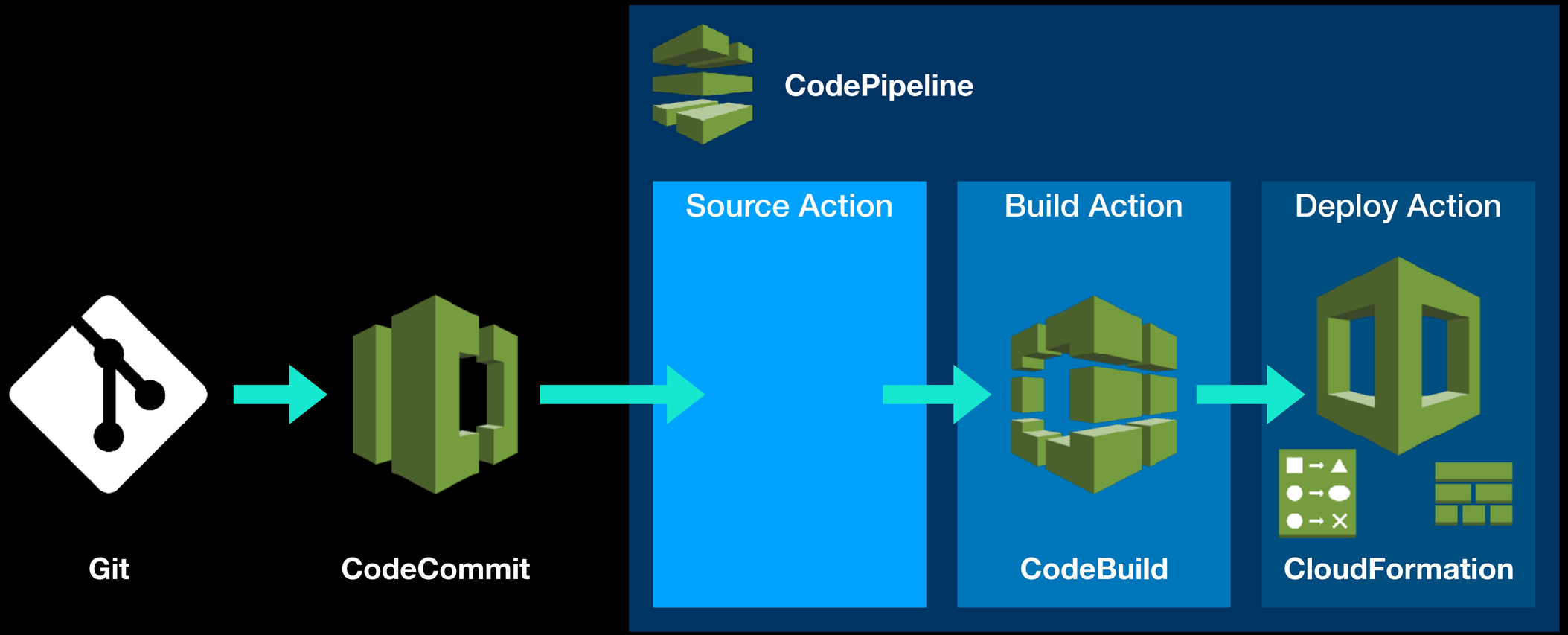
Mitigations

1. All builds start from scratch
   Yup.
2. Containers take time to instantiate and spin up
   Service-level caching
3. Pipeline components don't share state
   Yup.
4. Any state created during build is lost
   Build-level caching

Questions?

Tutorial

Tutorial Architecture



Overview

- Phase 1: Introduction to CodeCommit
- Phase 2: Introduction to CodeBuild
- Phase 3: Continuous Integration using CodePipeline
- Phase 4: Continuous Delivery using CloudFormation
- Phase 5: Speeding up CodeBuild

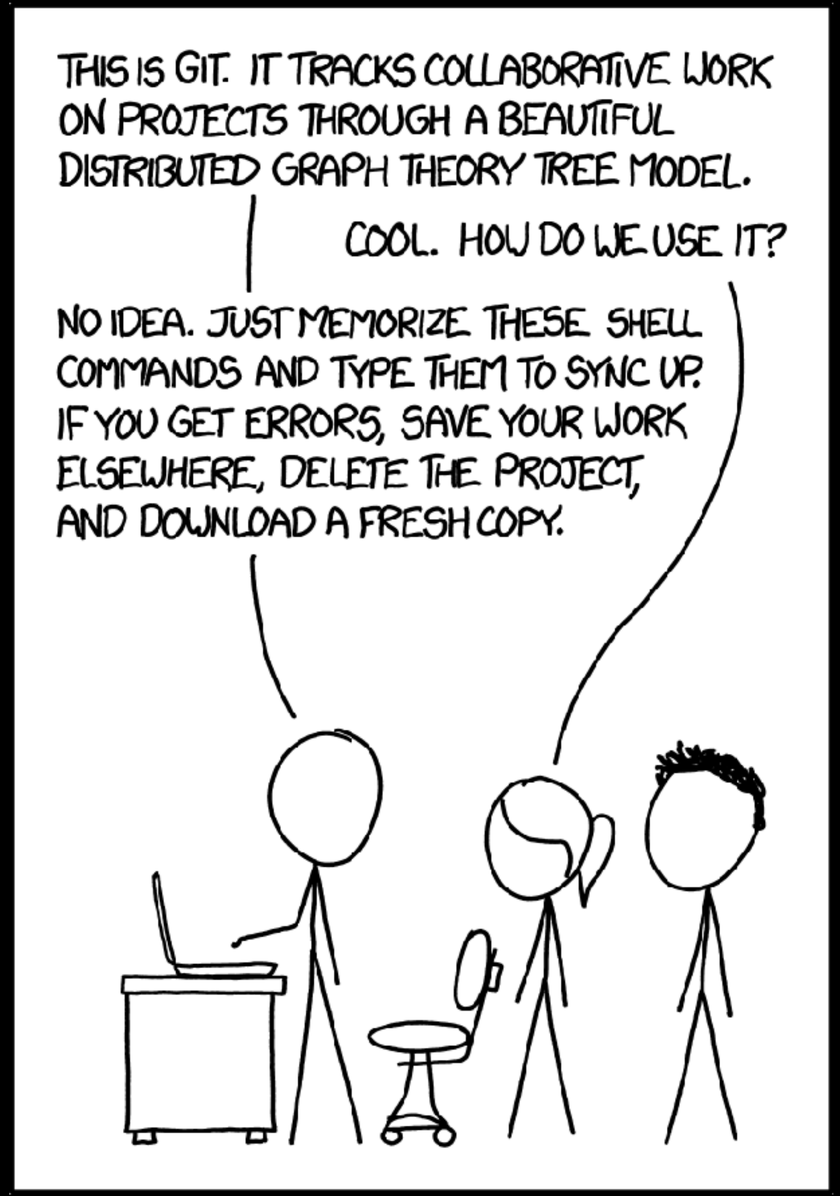
Logistics

- https://github.com/symphoniacloud/symphonia-sacon-nyc-2018
- Follow along, don't worry about keeping up
- Meet your neighbors
- Ask questions

Phase 1

Introduction to CodeCommit

Git and IAM, together at last...



Phase 2

Introduction to CodeBuild

Builds... can't someone else do it?

Phase 3

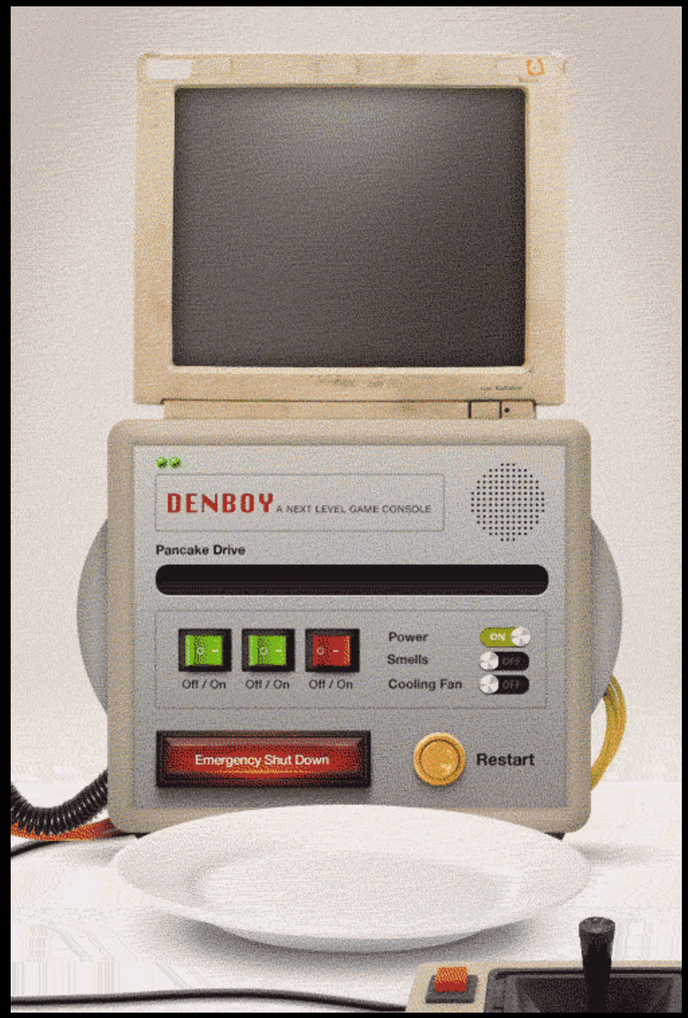
Continuous Integration

Extreme Programming, to the MAX!

Phase 4

Continuous Delivery

Like Continuous Integration, but for customer feedback.



Phase 5

Speeding up CodeBuild

Just press the Turbo button.

Teardown

- Delete all S3 buckets via the web console
- Delete CodeCommit repository
- Delete all CloudFormation stacks

Discussion and Questions

sacon@lists.symphonia.io

john@symphonia.io