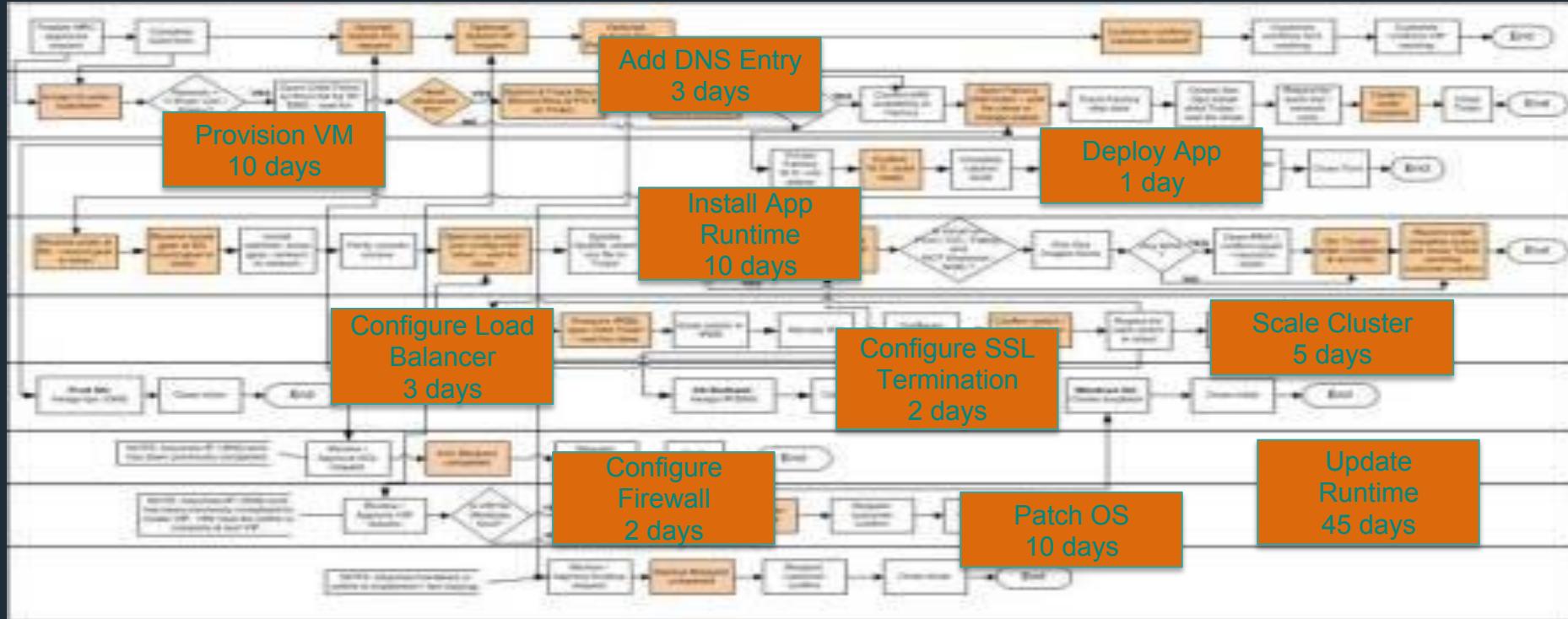


Cloud Native Enterprise

May 2016

Traditional IT

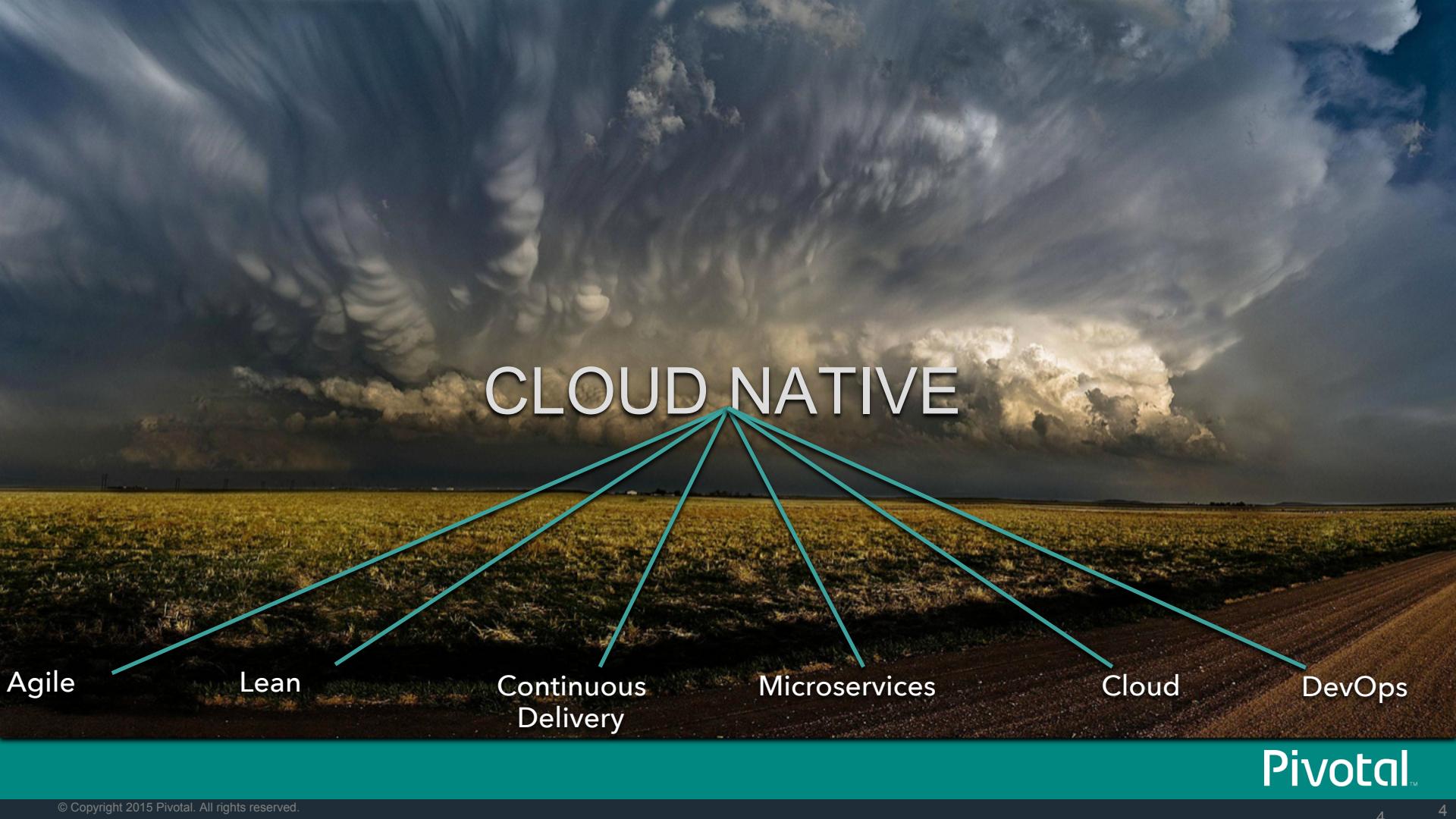


What makes these companies special?



NETFLIX





CLOUD NATIVE

Agile

Lean

Continuous
Delivery

Microservices

Cloud

DevOps

Pivotal™

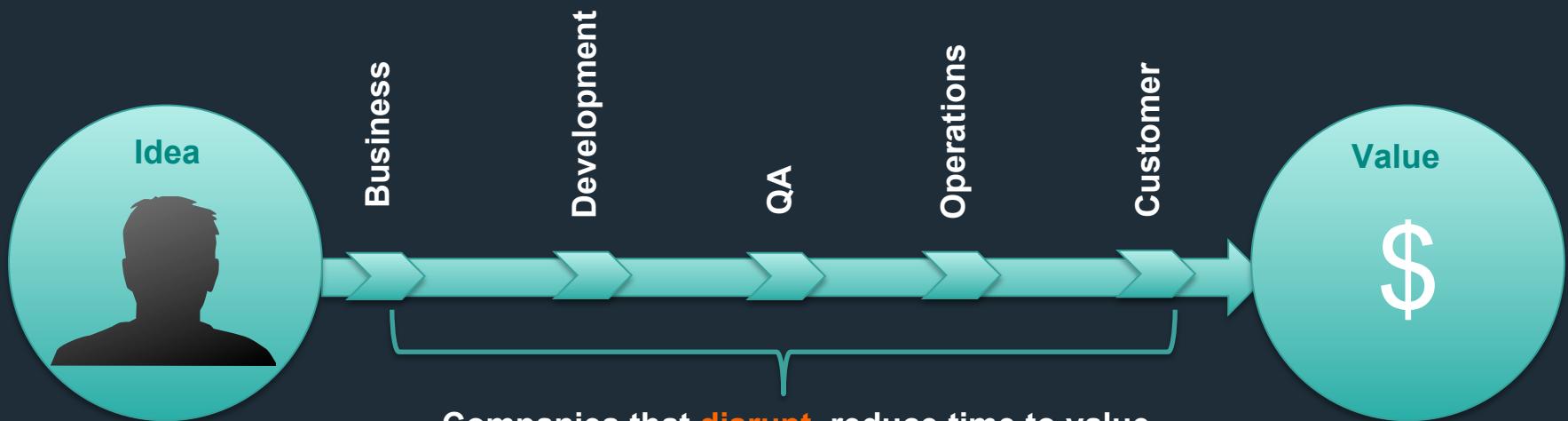
The promise of Cloud Native

- Deploy new features daily to production
- Automate middleware
- Free/Cheap horizontal scaling
- Contract between App & Platform

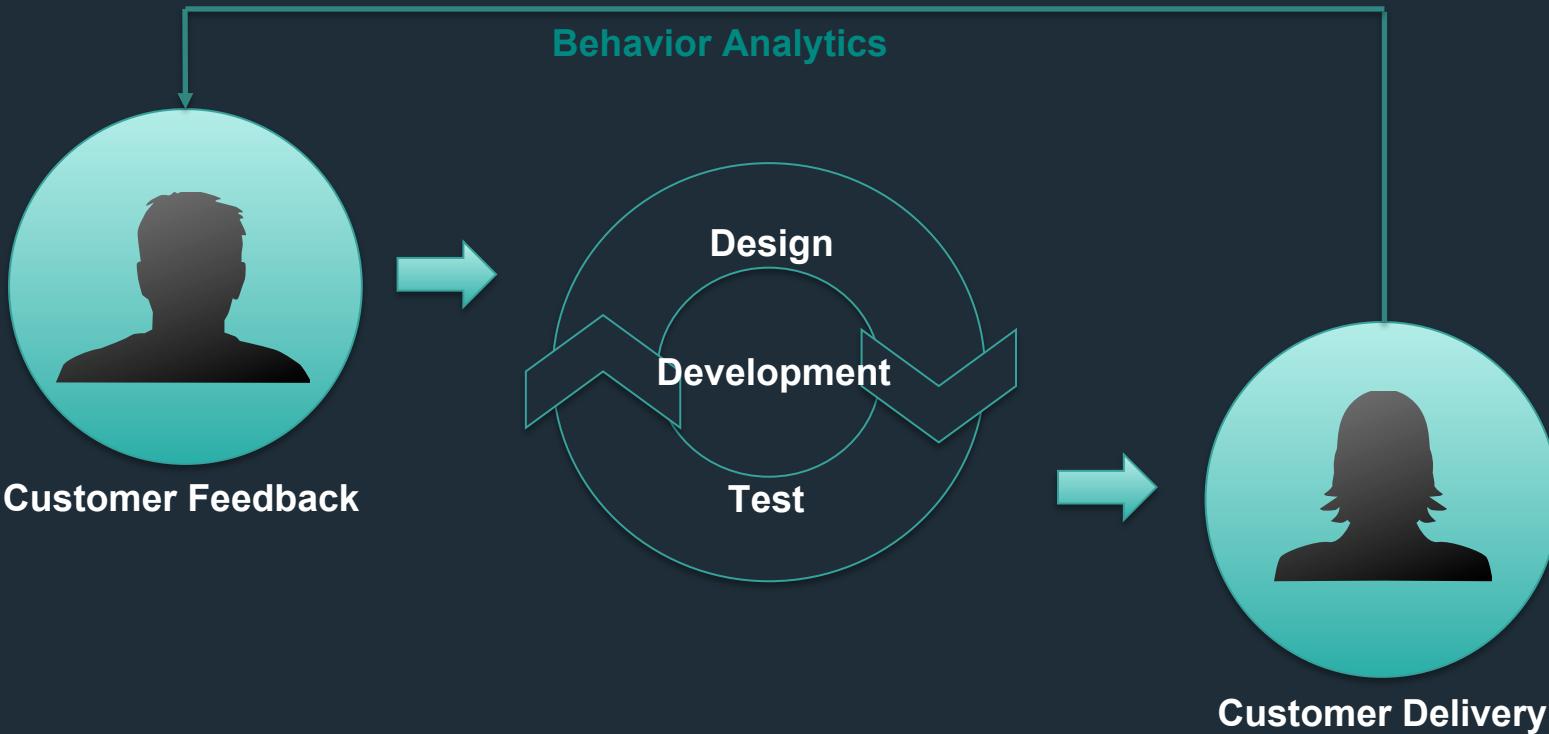
Characteristics of Cloud Native Architectures

- Continuously Delivered
- Twelve Factor Apps (<http://12factor.net/>)
- Microservices
- Self-Service agile infrastructure
- API-based collaboration
- Anti-fragility

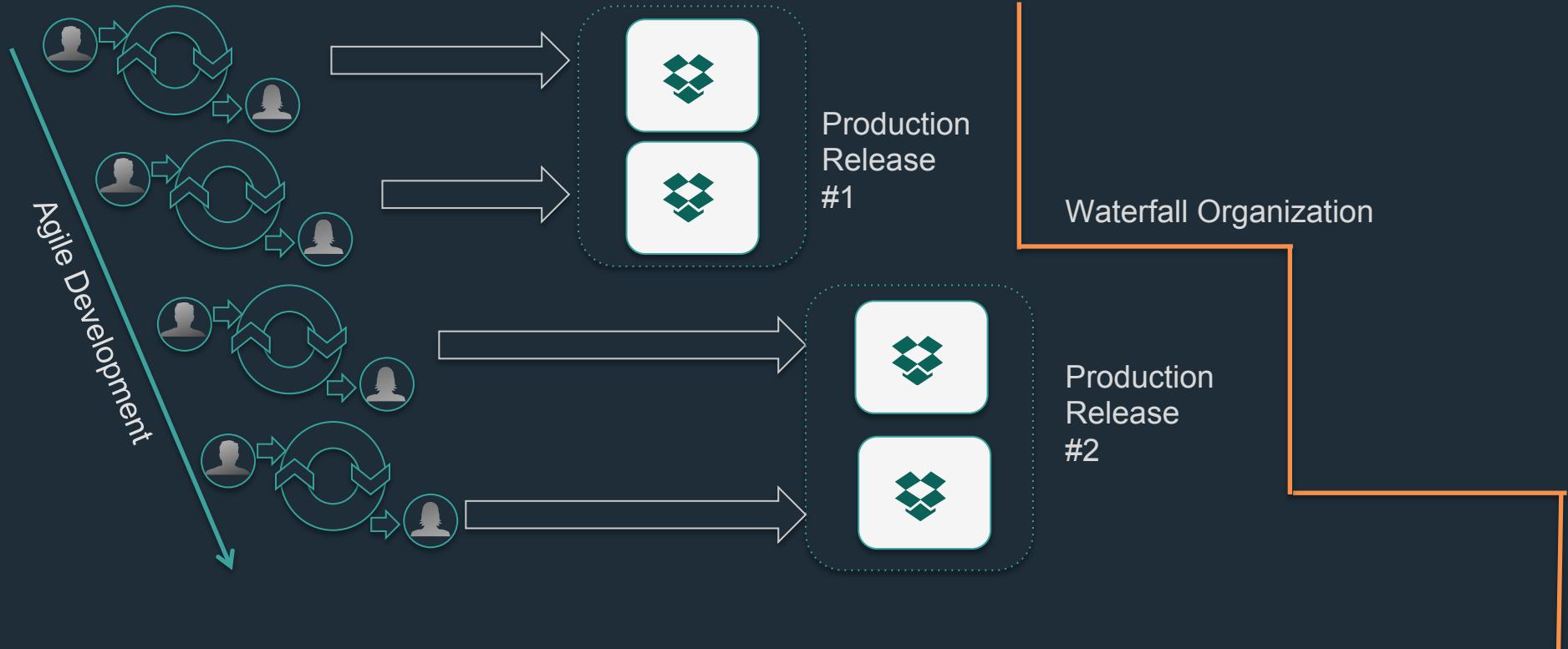
Enabling Continuous Delivery



Enabling Agile Delivery



WaterScrumFall



Conway's Law

“Any organization that designs a system (defined broadly) will produce a design whose structure is a copy of the organization's communication structure.”

- Melvin Conway, 1967

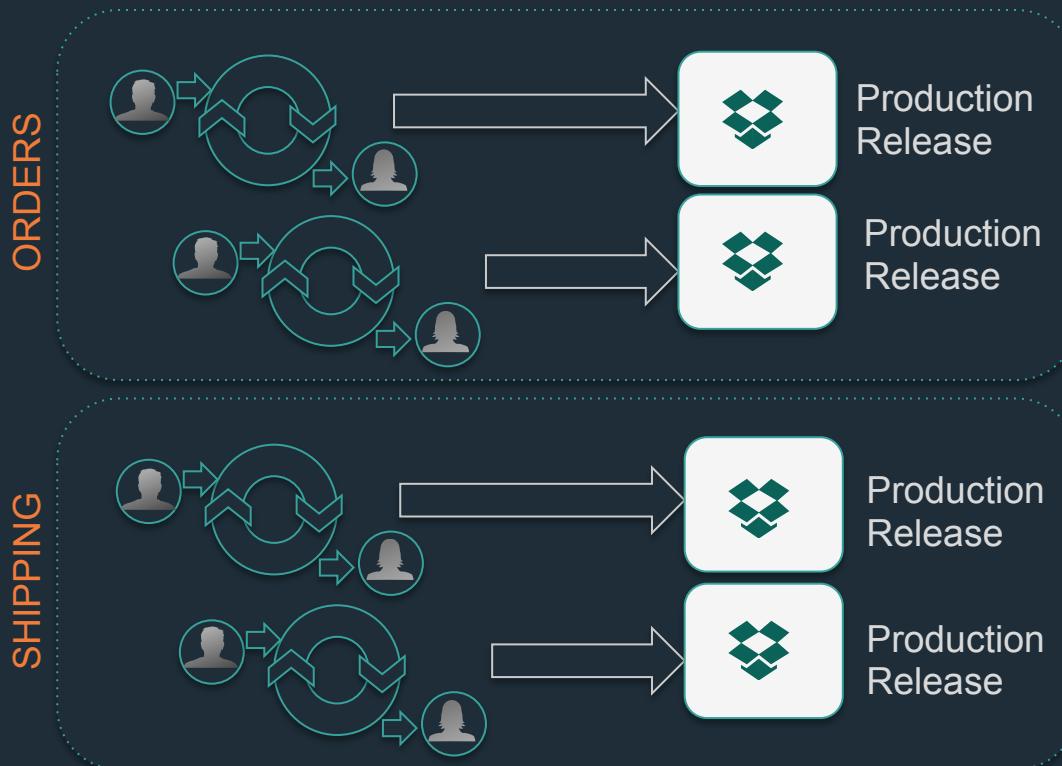
Inverse Conway Maneuver

Enable Continuous Delivery



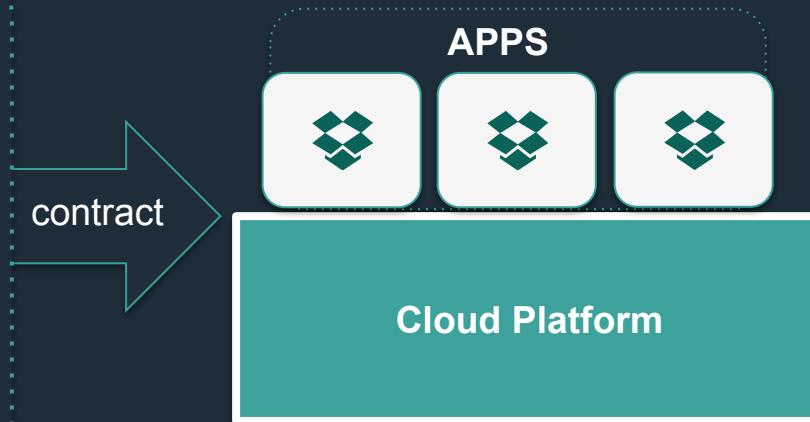
<http://jonnyleroy.com/2011/02/03/dealing-with-creaky-legacy-platforms/>
<http://www.slideshare.net/adriancockcroft/microservices-the-good-bad-and-the-ugly>

Enabling Continuous Delivery



Twelve-Factor Apps

- 1. Codebase
- 2. Dependencies
- 3. Configuration
- 4. Backing Services
- 5. Build, release, run
- 6. Processes
- 7. Port binding
- 8. Concurrency
- 9. Disposability
- 10. Dev/prod parity
- 11. Logs
- 12. Admin processes



Architectural and development practices – <http://12factor.net>

Microservice: Definition

If every service has to be updated in concert, it's not loosely coupled!

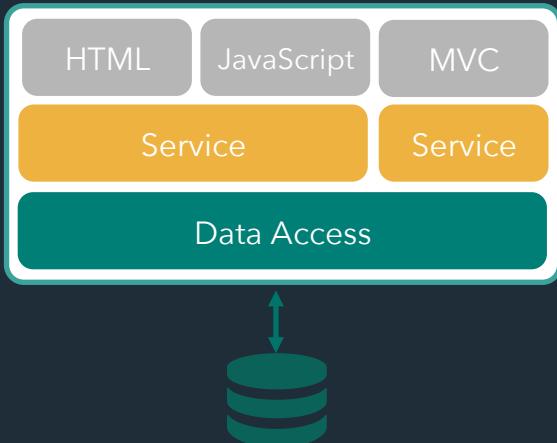
“Loosely coupled service oriented architecture with bounded contexts”

If you have to know about surrounding services you don't have a bounded context.

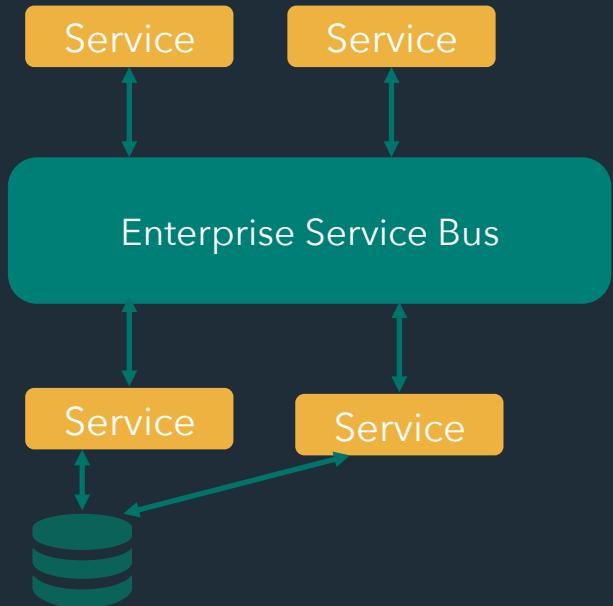
- Adrian Cockcroft

Microservices are NOT

Monolithic Application



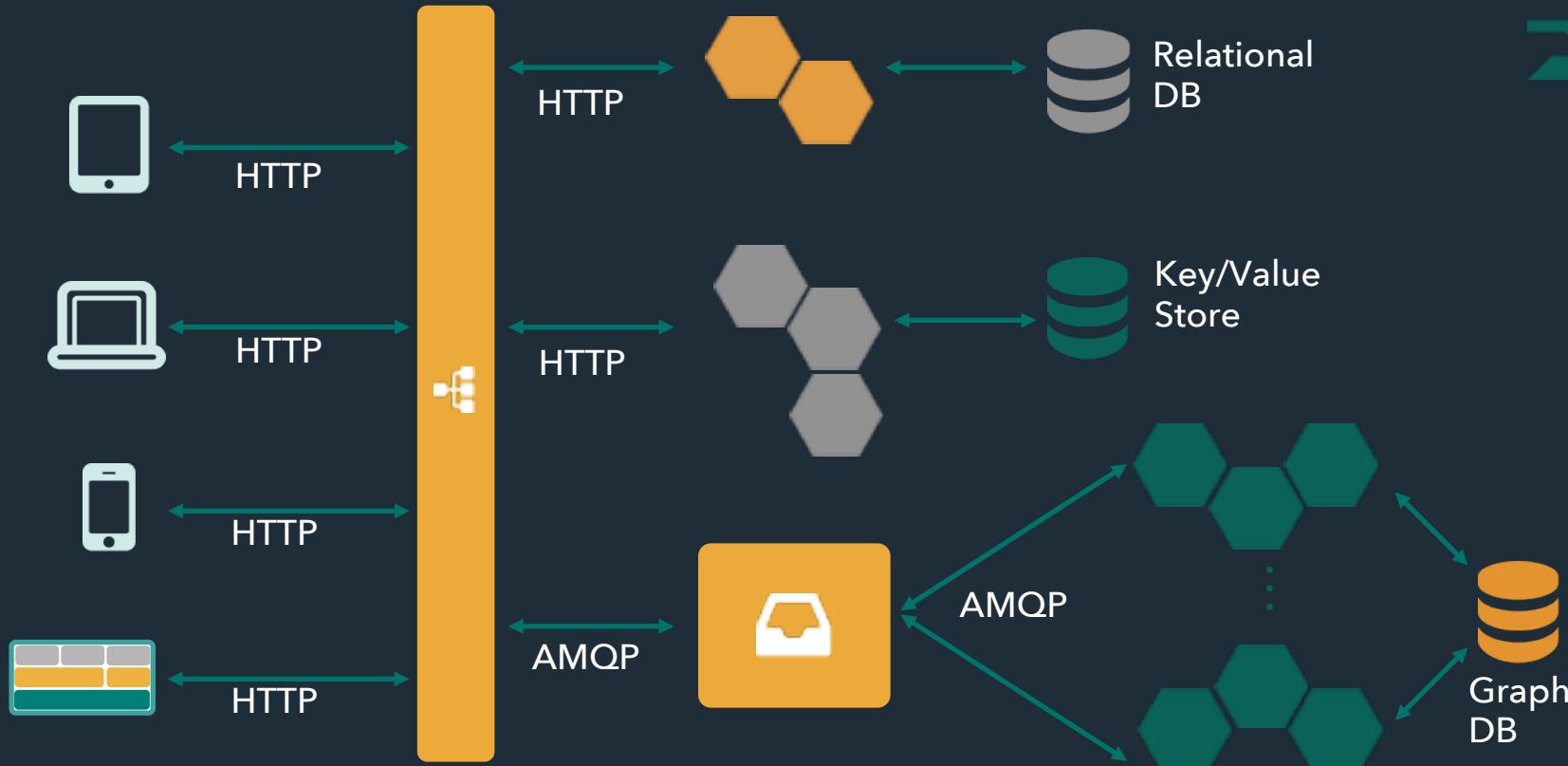
OR



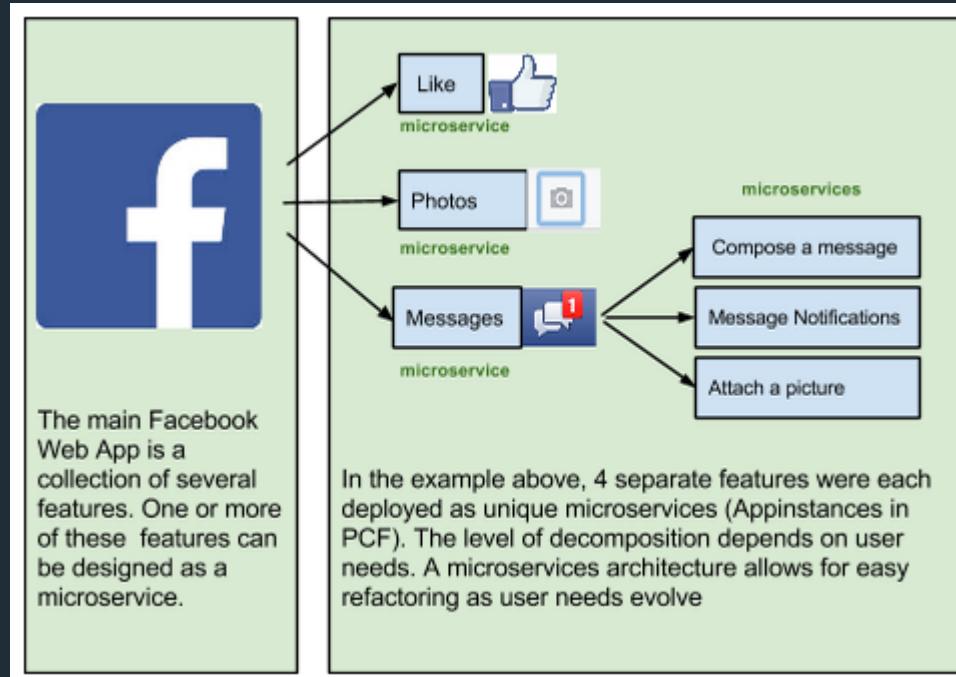
Tightly Coupled

Centralized

Microservice Architecture



A familiar Microservice example



Challenges of distributed systems

- Configuration Management
- Service Registration & Discovery
- Routing & Load Balancing
- Fault Tolerance (Circuit Breakers)
- Monitoring
- ...

Companies want to be **fast** like Netflix

- Netflix needed to be faster to win / disrupt
- Pioneer & vocal proponent of microservices - the key to their speed and success
- Netflix OSS supplies parts, but it's not a solution
- Difficult for enterprises to build it themselves
- Pivotal offers the closest thing to “Netflix in a box” today



“

“Velocity on the JVM is the Killer App”

- Andy Glover (Netflix Eng) @ SpringOne2GX 2014 Keynote

<https://youtu.be/xU267-YHN5c?t=1938>

It takes a platform

An end-to-end platform that
makes implementing
distributed application best
practices, a **turn-key** and **first**
practice

Anatomy of a Cloud Native Platform

Developer



1. Application Framework



Contract – 12 Factor App

Dev+Ops

2. Container Runtime



Contract – BOSH Release

IT Ops

3. Infrastructure Automation



Contract – Cloud Provider Interface

IT Ops

4. Infrastructure



“Contracts” between Applications, opinionated frameworks like Spring Boot and Spring Cloud and opinionated Cloud Native Platforms like Cloud Foundry help significantly accelerate the development of Cloud Native applications



Application Framework



Spring Cloud



Spring Boot



Steel Toe

Spring Boot



From 0 to app in < 5 min

Spring Cloud



Designed for fragile infrastructure in
partnership with **Netflix**

Steel Toe



Bringing Spring goodness to .Net with Microsoft

Pivotal™

Spring Cloud Services Suite



Spring Cloud
Services



Config Server

- Git URL for Config Repo provided via Service Dashboard (post-provisioning)
- Single tenant, scoped to CF space



Service Registry

- Service Registration and Discovery via Netflix OSS Eureka
- Registration via CF Route



Circuit Breaker
Dashboard

- Netflix OSS Turbine + Hystrix Dashboard
- Aggregation via AMQP (RabbitMQ)

Self service

Auto scaling

Container management



Container Runtime



Cloud Foundry

Self healing

Integrated logging

**Application Performance
Management**

Pivotal™

**manage services
not servers**

**no maintenance
windows**

zero downtime

embedded OS

infra as code

multi cloud support



IT Ops

Infrastructure Automation



IT Ops

Infrastructure



BOSH



AWS



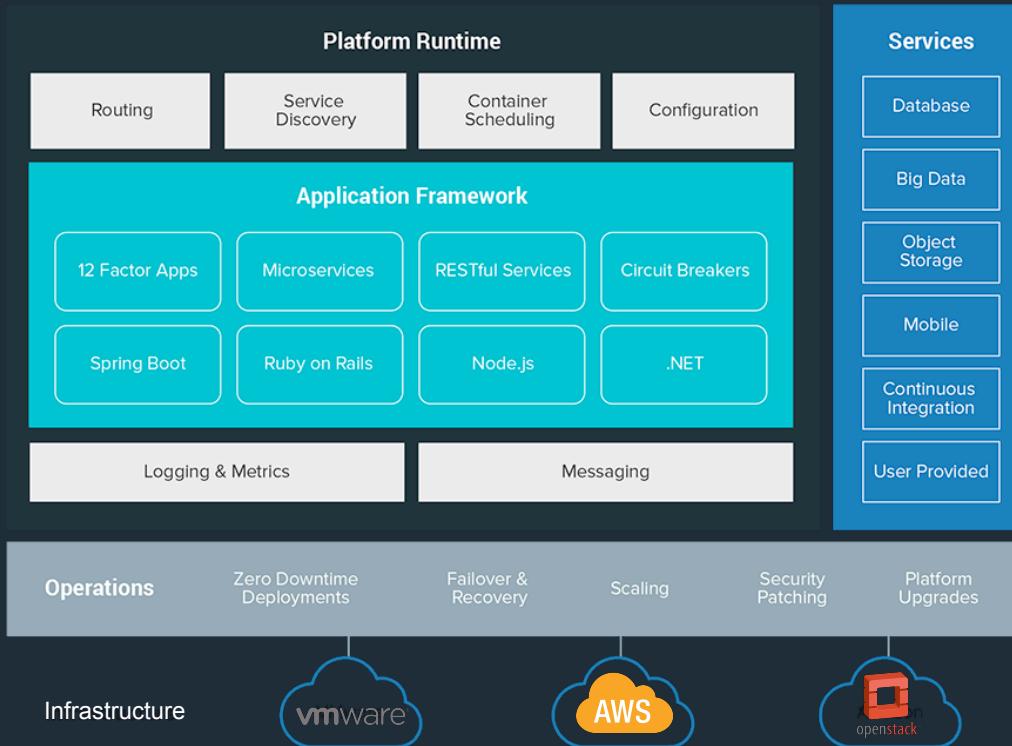
VMWare



OpenStack

Pivotal

It Takes a Platform



Cloud Native Maturity Model

Cloud Native

- Microservices architecture
- API-first design

Cloud Resilient

- Fault-tolerant and resilient design
- Cloud-agnostic runtime implementation
- Bundled metrics and monitoring
- Proactive failure testing

Cloud Friendly

- 12 Factor App methodology
- Horizontally scalable
- Leverages platform for high availability

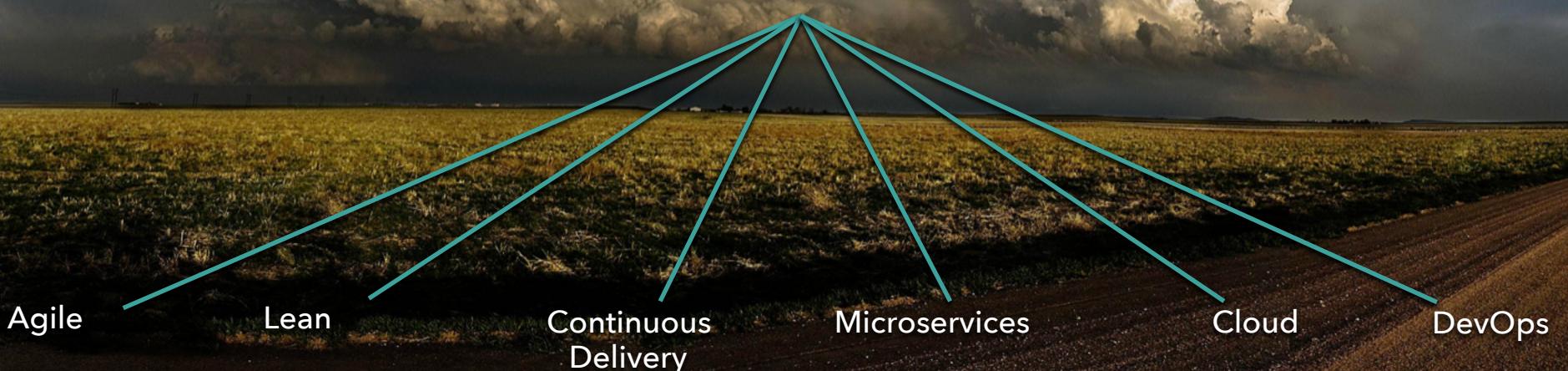
Cloud Ready

- No permanent disk access
- Self-contained application
- Platform-managed ports and networking
- Consumes platform-managed backing services





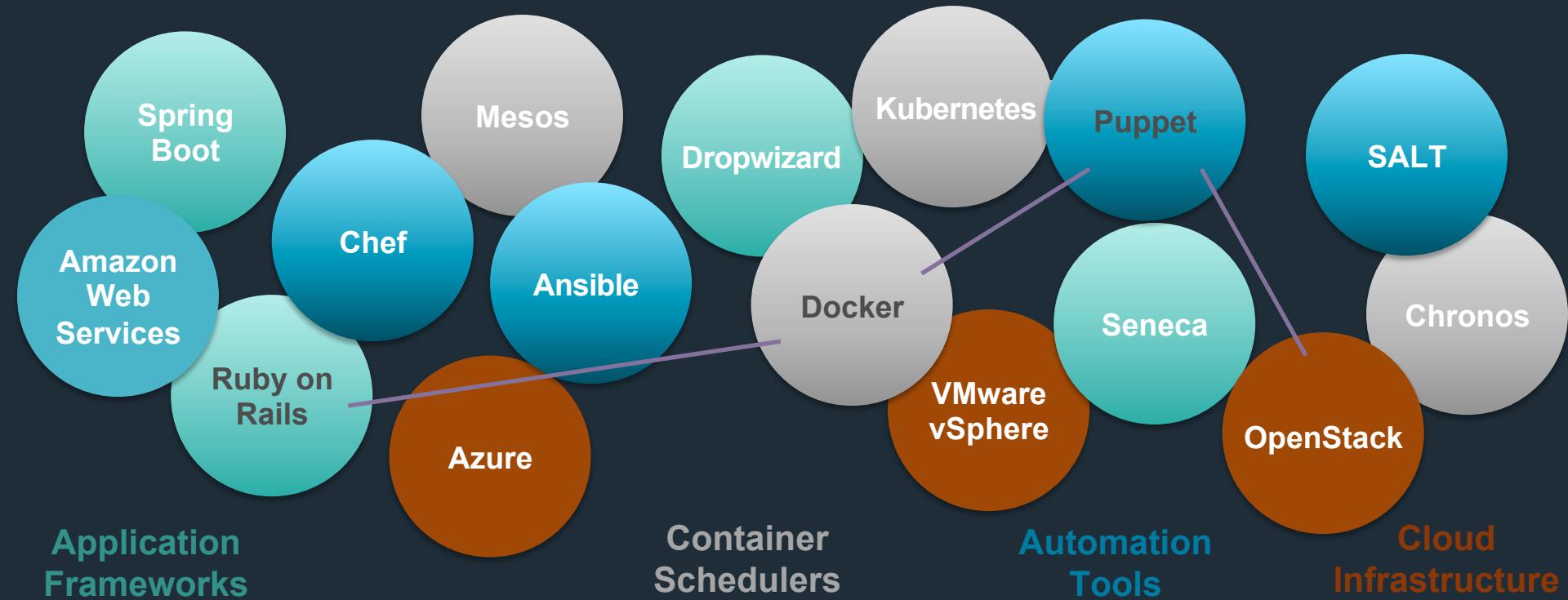
What's CLOUD NATIVE for you?



Pivotal™

Don't Play “Cloud Technology Bingo”

Cloud adoption with non-cloud native applications, unopinionated tools and adhoc automation



Application
Frameworks

Container
Schedulers

Automation
Tools

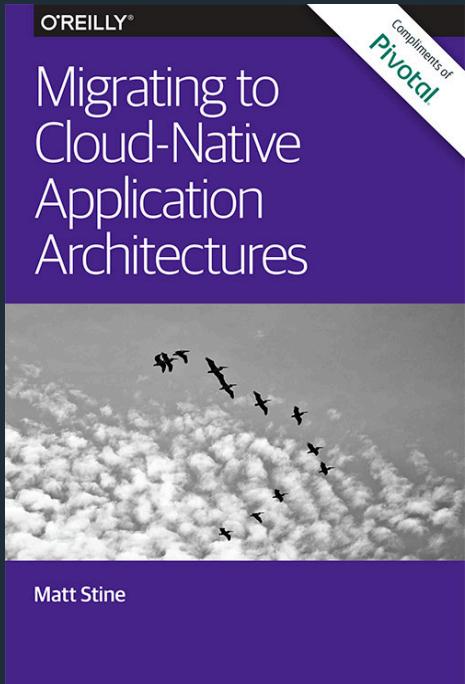
Cloud
Infrastructure

Pivotal™

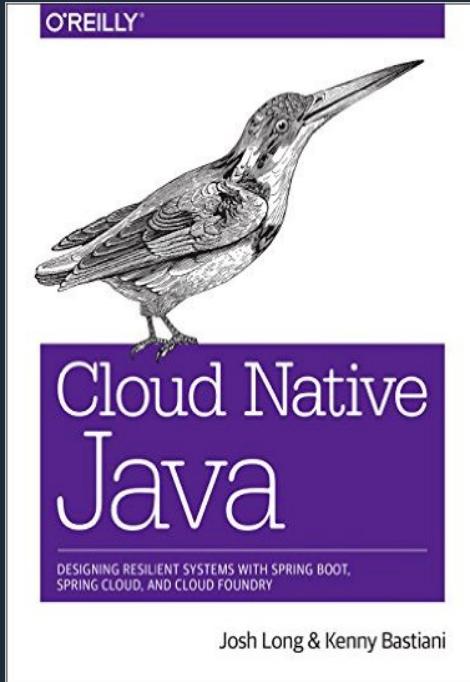
Pivotal

A NEW PLATFORM FOR A NEW ERA

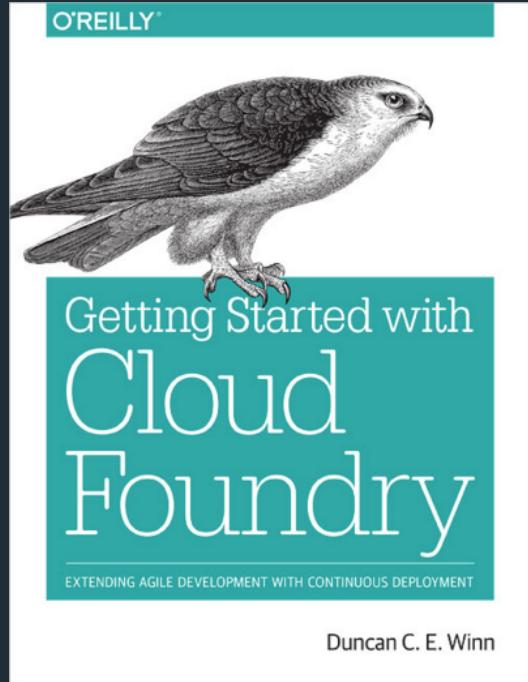
Read



By Matt Stine (@mstine)



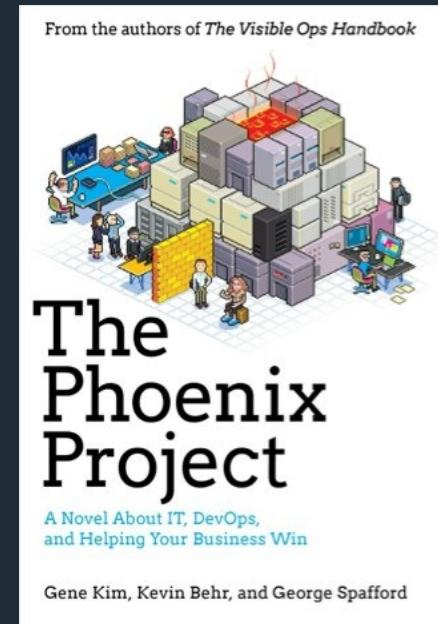
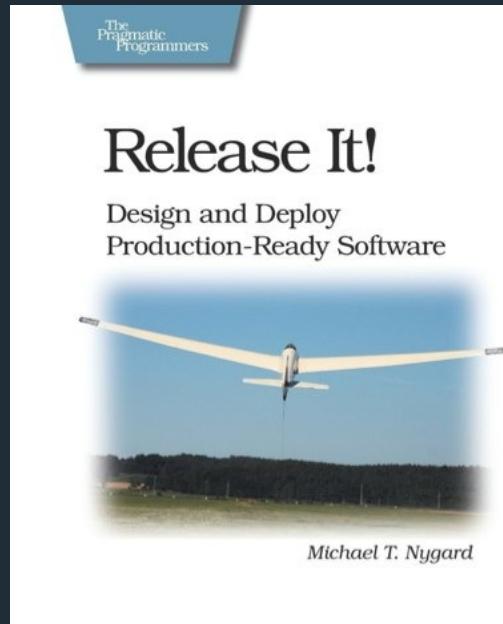
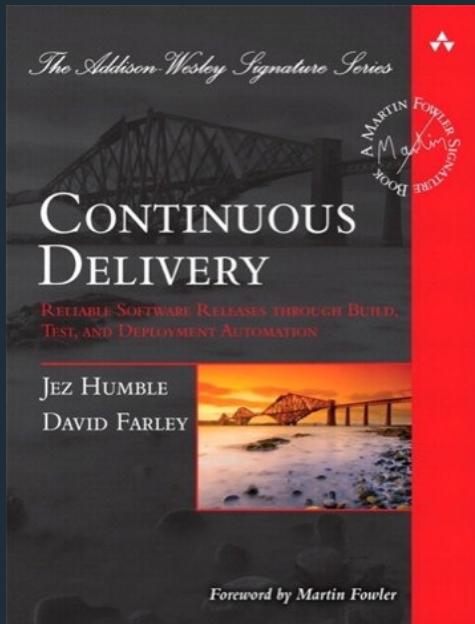
By Josh Long (@starbuxman)
and Kenny Bastani



By Duncan Winn (@duncwinn)

Pivotal™

Foundations



Foundations

