

Agenda

- Workshop Kick-Off
- Introduction to Pivotal Cloud Foundry
- Labs
- Labs Review
- Cloud Native Design and Microservices
- Some More Labs
- Spring Cloud Services & Steeltoe (Netflix OSS for .NET)
- Demo (or Lab)
- Q&A

Pivotal Cloud Foundry

Introduction

WE BELIEVE

Transforming how the world builds software
will shape the future.

From idea to production: 6-9 months



An idea in the morning, is running in production by evening



Software is changing INDUSTRIES



Square

\$6 BN

Financial Services



\$53 BN

Entertainment



\$26 BN

Hotel



\$34 BN

Automotive



U B E R

\$50 BN

Transportation



\$3.2 BN

Industrial Products

The Enterprise **REVOLUTION**



700+ apps



Spring + PCF



Re-writing software
the modern way



Agile software
transformation

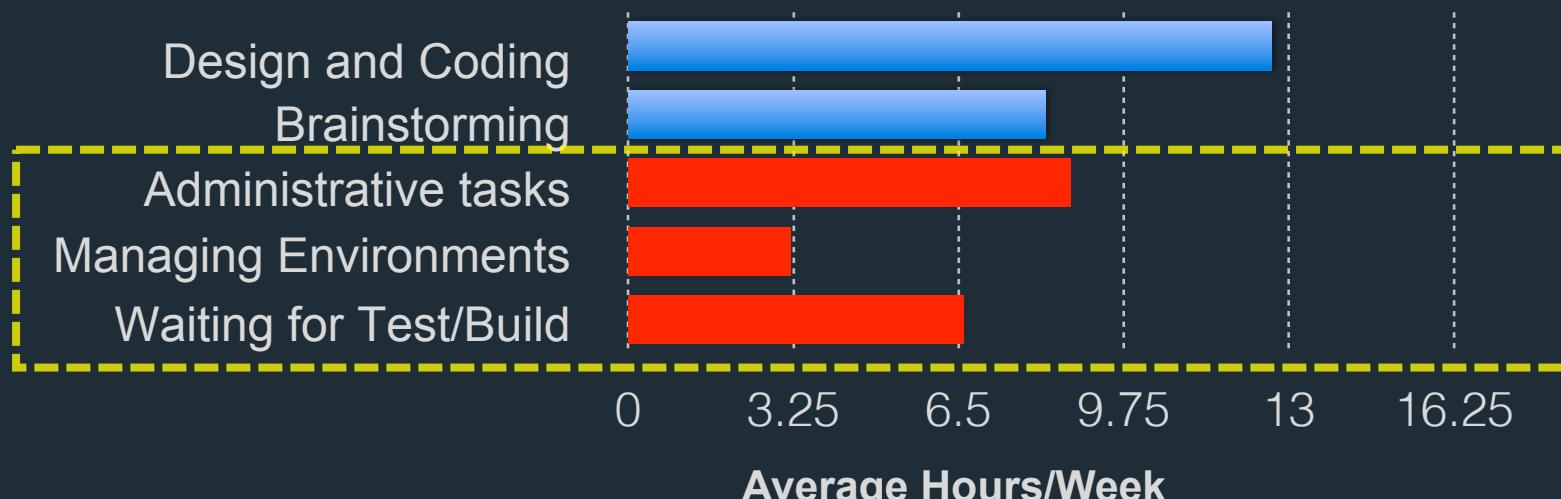


DevOps adoption with PCF
Automated build pipeline



Major IT
transformation

Software Developers Spend too Much Time NOT Writing Software



17.5 hrs ~ 40% of full work week!

Complex

The Development Team's Duties

- Understand Requirements
- Assess Impact
- Provide Estimates
- Attend Meetings
- Design Solutions
- Write Code
- Build Code
- Unit Test Code
- Reports Status
- Peer Review Code
- Integrate Code
- System/Integration Testing
- Create and submit QA eTP requests
- Create and submit Change Orders for QA
- Verify/Smoke Test QA turns
- Research Defects
- Resolve Defects
- Create and submit PROD eTP requests
(document deployment risks and plans with turn)
- Create and submit Change Orders for PROD
- Attend CAB and/or A-CAB
- Verify/Smoke Test PROD turns
- Closeout after business signoff

Only one of these
is writing code!

Directional Summary

Current

- Artifact driven process
- Project-centric
- Inspection/Test focused quality controls
- Development teams focused solely on project requirements
- External gates requiring extra steps to “prove” worthiness
- Ops/PSM transition varies from team-to-team



Future

- Process and tools create artifacts
- Product/Service-centric
- Empowered development teams (Knowledge Workers) aligned to the bigger picture
- Quality activities in all phases
- Metrics and tool-generated artifacts that demonstrate quality prior to deployment
- Operational Impact and Readiness integrated into the whole project lifecycle.

The developer dream haiku

Here is my source code,

Run it in the Cloud for me,

I do not care how

Deploying Code Shouldn't be So Painful (.NET)

Traditional

- Provision a VM, IP, DNS
- Install IIS
- Install .NET FWK
- Deploy Application
- Configure Load Balancer
- Configure SSL Termination
- Configure Firewall
- Configure Monitoring
- Configure Logging

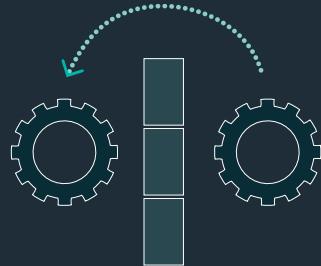
Pivotal Cloud Foundry

`cf push`

DevOps

Not my problem

Separate tools, varied incentives,
opaque process



Shared responsibility

Common incentives, tools,
process and culture



Continuous Delivery

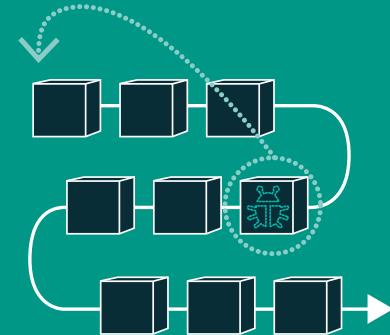
Release once every 6 months

More Bugs in production



Release early and often

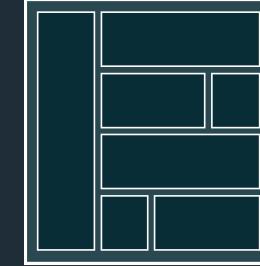
Higher Quality of Code



Microservices

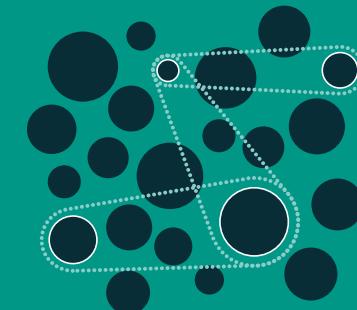
Tightly coupled components

Slow deployment cycles



Loosely coupled components

Automated deployments



Then

assume **reliable** infrastructure
release code **every 3 months**
works in **my** environment
tightly coupled

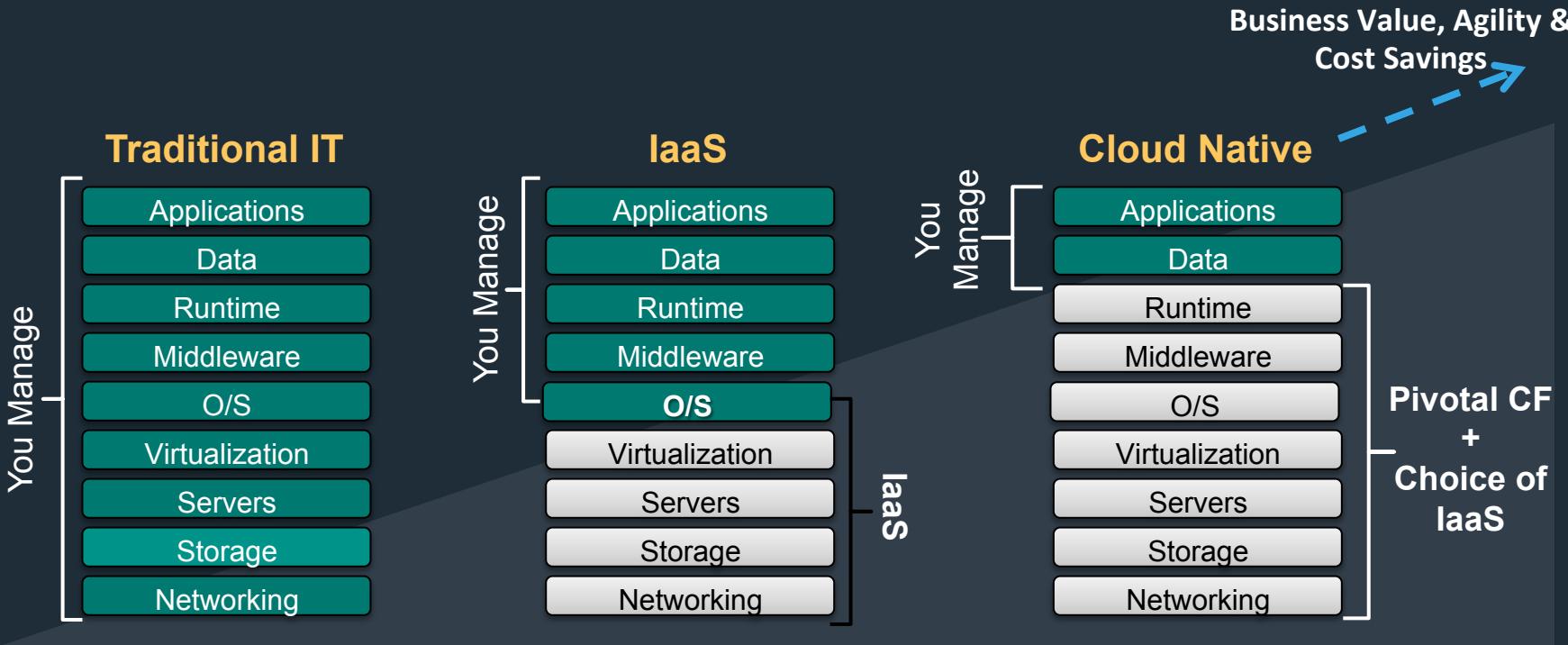
Now

assume **fragile** infrastructure
release code **early and often**
shared Dev & Ops responsibility
loosely coupled

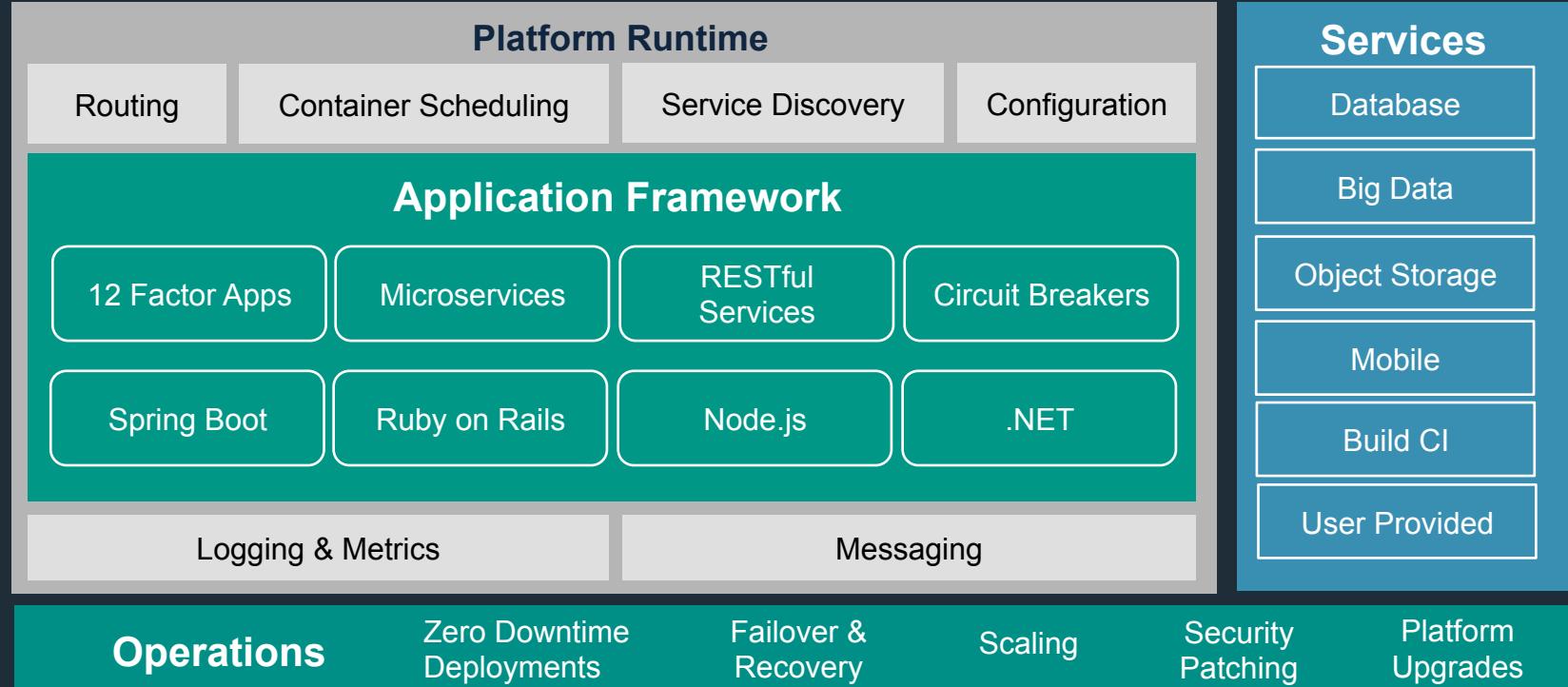
It Takes a Platform

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

Application-centric Platforms



Pivotal Cloud Foundry at a Glance



Everything Needed to Deploy and Operate an App



The image displays two screenshots of Pivotal software interfaces. The left screenshot shows the "Service Registry" interface with sections for "Service Registry Status" (listing "FORTUNES" under "Registered Apps" and "default (1)" under "Availability Zones") and "System Status" (showing "Parameter", "Current time", "Lease expiration enabled", and "Self-preservation mode enabled"). The right screenshot shows the "Circuit Breaker Dashboard" for the "Circuit" and "Thread Pools". The "Circuit" section includes a line graph and metrics for "ui.random Fortune" (Mean: 0.0/s, Closer: 0.0/s, Circuit Closed). The "Thread Pools" section lists "Fortuneservice" with metrics for "Active", "Queued", and "Pool Size". The bottom of the dashboard has a copyright notice: "Pivotal © 2015 Pivotal Software Inc. All rights reserved."



Pivotal
Cloud Foundry®



Microsoft Azure

Momentum for Digital Transformation with Cloud Native Microservices



spring



Spring Boot

Q2 2015
Public Preview of OSS
Cloud Foundry

Q4 2015
Public Preview of PCF

Q1 2016
Pivotal-managed PCF
via Azure Marketplace

Q2 2016
Microsoft, Ford Series C
investment in Pivotal

November, 2016
Fully automated Pivotal Cloud
Foundry with Ops Manager



and more...

Pivotal

PCF on Azure Co-Investment Pillars

Azure as the best IaaS for PCF



- PCF on Azure GA
- Tight integration with Azure IaaS and PaaS services with Service Brokers and Tiles
- Easier installation of PCF via Azure Marketplace
- Integrating Azure services into PCF Internals
- OMS / Active Directory/ API Management integrations

.NET & Java collaboration



- Windows Stemcells
- .NET Core BuildPack GA
- SteelToe RC
- VSTS & VS integrations for .NET development
- Java collaborations on Spring Data Flow, Spring Boot

Cloud Foundry Fundamentals



- PCF Support integration
- Continued investment in Azure CPI
- Reference Architectures for key verticals

Azure Stack

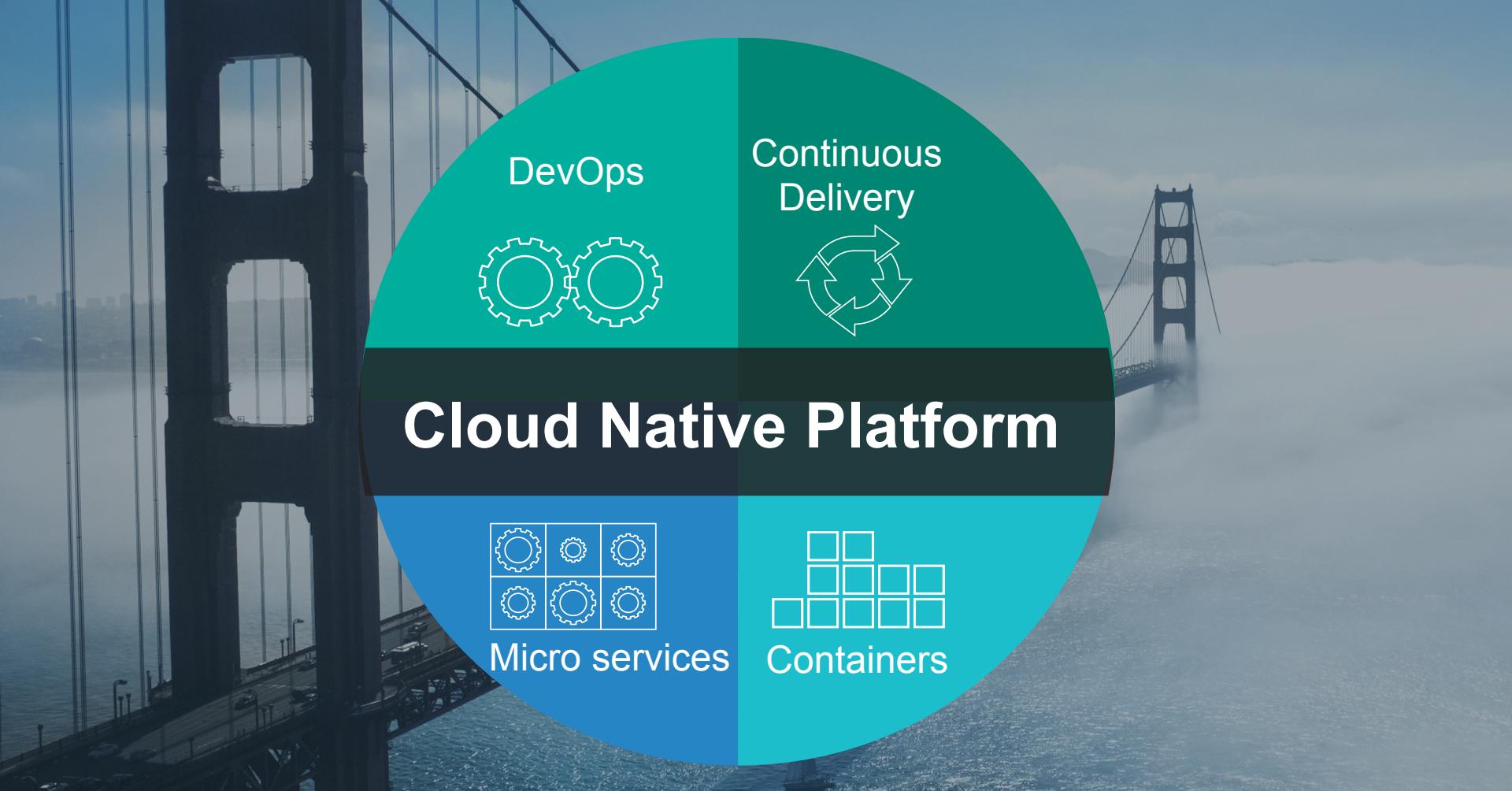


- CPI Testing
- PCF at GA

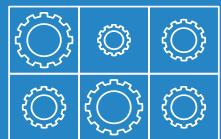
Cloud is about
how computing is
done, not where.

-Paul Maritz

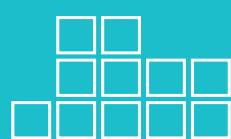




Cloud Native Platform



Micro services



Containers



BREAK