



CLOUD NATIVE APPLICATIONS CONTAINERS, MICROSERVICES, PLATFORMS, CI-CD... OH MY!!

FABIO CHIODINI

Principal System Engineer

@FabioChiodini

WHY ARE WE HERE?



A word cloud illustrating various concepts in cloud computing and DevOps. The words are arranged in a dense, overlapping manner, with colors ranging from dark blue to light yellow. The most prominent words include:

- Unstructured CD
- Platform App CI
- Containers
- Microservices
- Deployment
- Structured
- Cloud
- Native
- Integration
- Continuous
- Development
- Infrastructure
- Build
- Runtime
- VM
- Mesos
- Photon
- Delivery
- OpenStack
- Books
- Layers
- Culture
- Provisioning
- Service
- Apps
- EMC
- VxRack
- KVM
- Hardware
- Orchestration
- Automation
- Neutrino
- Docker
- DevOps
- Twitter
- PaaS
- Stack
- Factor
- Kubernetes
- Deploy
- IaaS

SOFTWARE IS TRANSFORMING INDUSTRIES

NETFLIX

ENTERTAINMENT

\$53BN

 Square

FINANCIAL SERVICES

\$6BN

 **airbnb**

HOTEL

\$26BN

T E S L A

AUTOMOTIVE

\$34BN

U B E R

TRANSPORTATION

\$50BN

nest

INDUSTRIAL PRODUCTS

\$3.2BN

ENTERPRISES ARE FOLLOWING



700+ apps



Spring Framework +
Pivotal CF as a Cloud
Native Platform



Re-writing software
the modern way



Agile software
transformation



DevOps adoption
with Pivotal CF
automated
build pipeline



Major IT
transformation



Mercedes-Benz

Connected cars and
Smart Apps



Bosch IoT Suite

CLOUD NATIVE APPLICATIONS

“Applications that do not require resilient infrastructure”

Cloud native is a term describing software designed to run and scale reliably and predictably on top of potentially unreliable cloud-based infrastructure.

Cloud-native applications are purposefully designed to be infrastructure unaware, meaning they are decoupled from infrastructure and free to move as required.*



Joshua McKenty @jmckenty · 30 Nov 2015

Cloud Native: Declarative and durable apps, built on 12 Factor principles and run using Platform. API first, use microservices to scale team



Cloud Native Apps: The modern, modular approach to building applications that are focused on web-scale, mobile first and real-time data.

*Duncan C.E. Winn

HOW DO THE TERMS FIT IN YOUR LIFE-CYCLE?

DevOps

NOT MY PROBLEM

Separate tools, varied incentives, opaque process



SHARED RESPONSIBILITY

Common incentives, tools, process, and culture



CI/CD

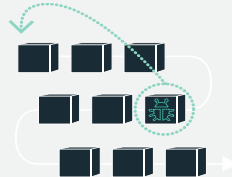
RELEASE ONCE EVERY 6 MONTHS

More bugs in production



RELEASE EARLY AND OFTEN

Higher quality of code



Microservices

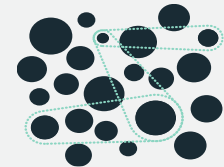
TIGHTLY COUPLED COMPONENTS

Slow deployment cycles waiting on integrated test teams



LOOSELY COUPLED COMPONENTS

Automated deploy without waiting on individual components



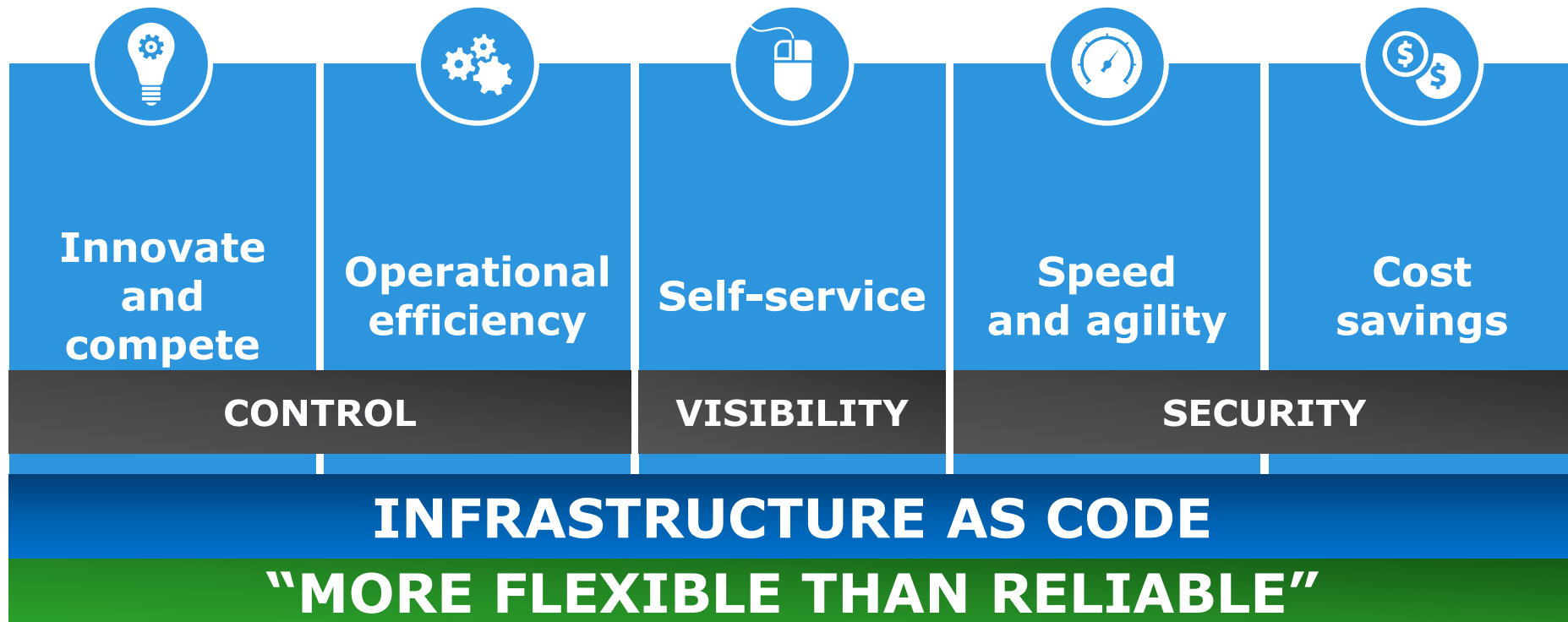
MANAGE

DEPLOY

DESIGN

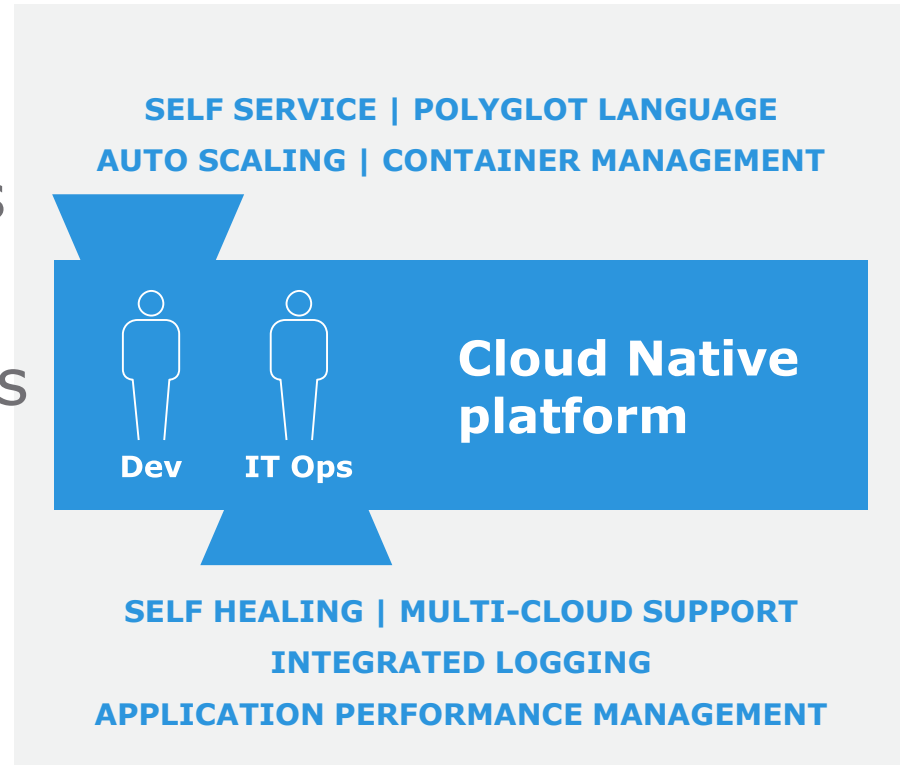
RESULTING IN NEW REQUIREMENTS FOR IT

TO DEPLOY AND DELIVER APPLICATIONS RELIABLY, AT SCALE



CLOUD NATIVE PLATFORM REQUIREMENTS

- **Programmability**
("Infrastructure As Code")
- **Elasticity** (Which Demands A Scale-Out Architecture)
- **Economics** (Steers Towards Standard Servers + Software)
- **Strong Instrumentation And Telemetry** Of Infrastructure Layer

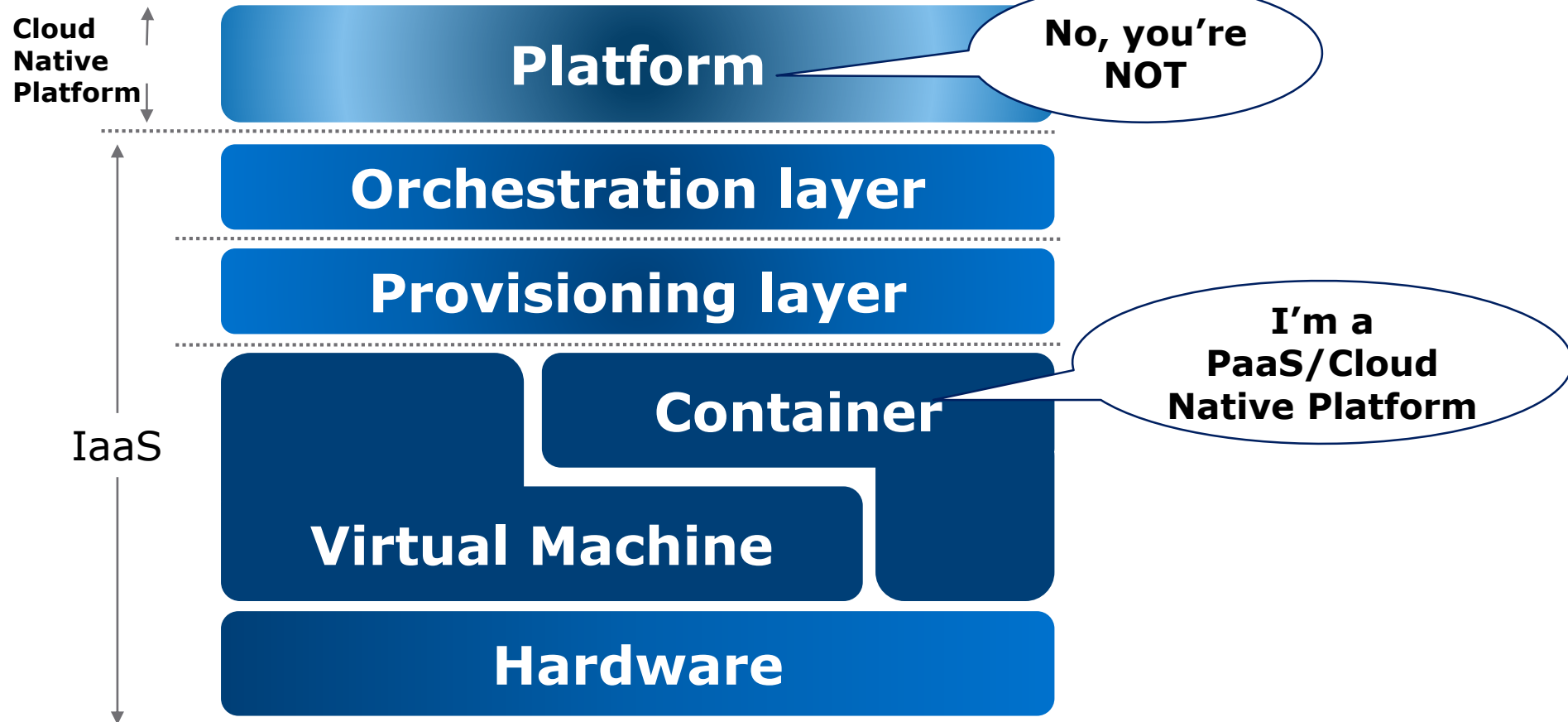


BTW...

CONTAINERS vs VMs

101

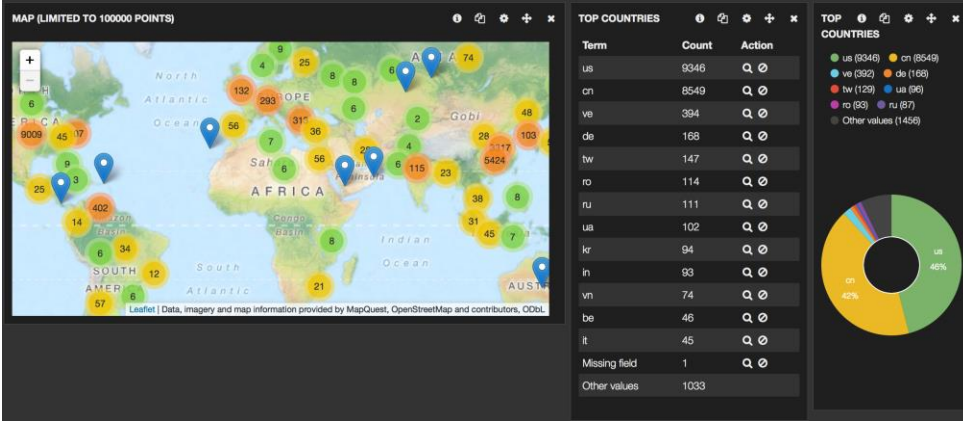
CLOUD NATIVE PLATFORM, IAAS AND CONTAINERS



Re-adapted from @swardley (Simon Wardley)

DEMO #1
POWER OF CNAs
[AKA CNP DO-IT-YOURSELF]

AN APP STORY... IN MICROSERVICES



A **Honeypot** is a computer security mechanism set to detect, deflect, or, in some manner, counteract attempts at unauthorized use of information systems

Honeypots Provisioning



Honeypot #1
(honeypot.py)



Honeypot #2
(honeypot.py)

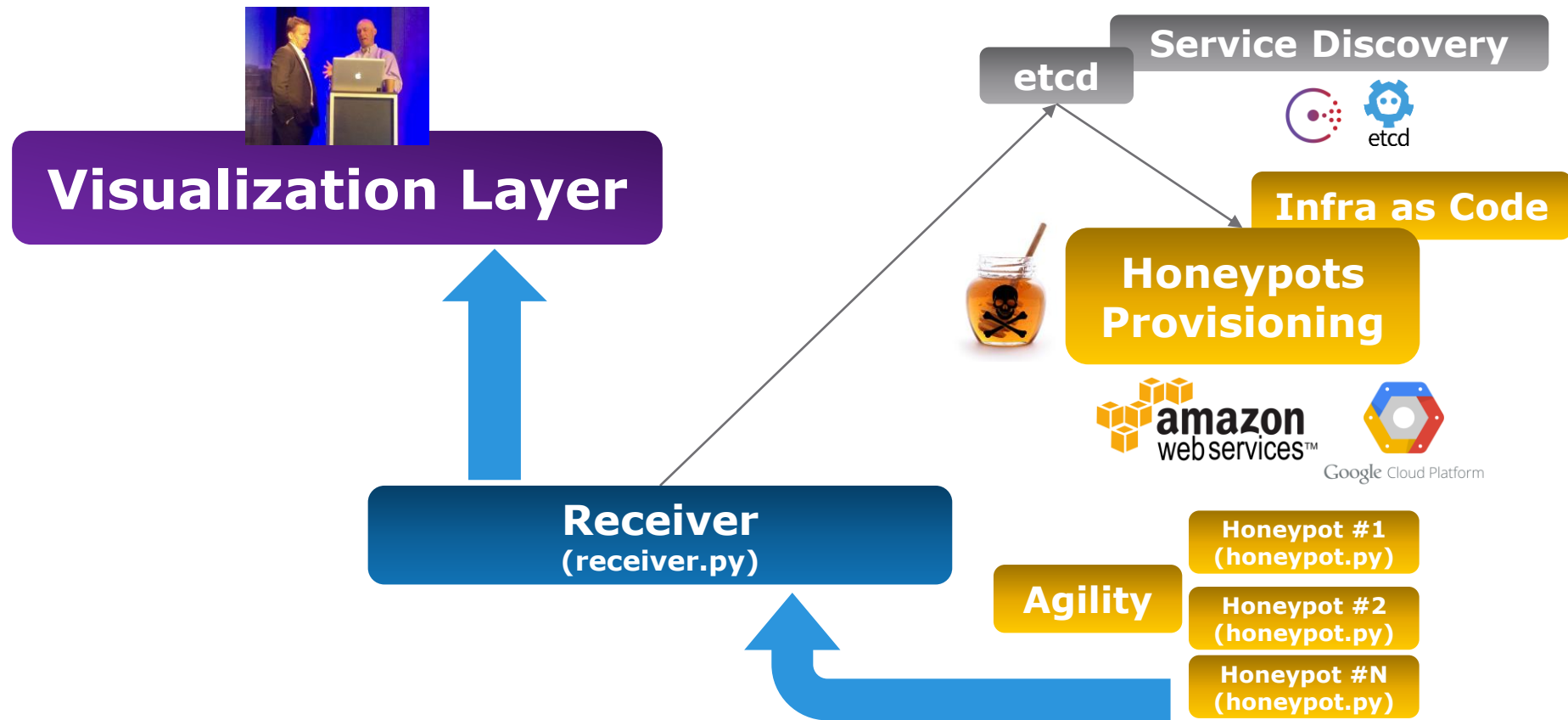


Honeypot #N
(honeypot.py)

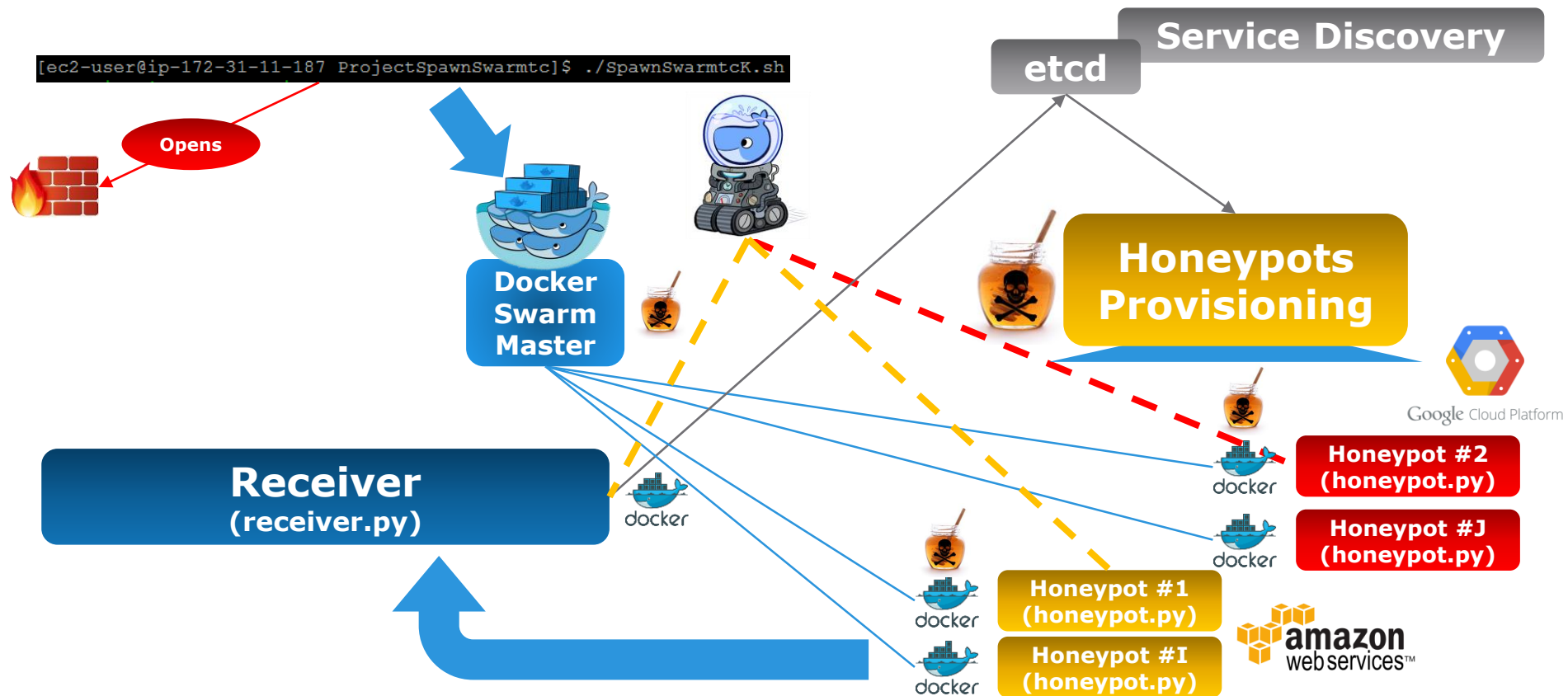
Receiver
(receiver.py)

Persistency

AN APP STORY... OVER MANY CLOUDS + TOOLS



AN APP STORY... WITH MANY TOOLS



IN OTHER WORDS...

GOAL: Continuous Delivery

WHAT: Cloud Native Apps

HOW: Cloud Native Platform

THROUGH: DevOps Culture

IN OTHER TWEETS



Joshua McKenty @jmckenty · 30 Nov 2015

Cloud Native: Declarative and durable apps, built on 12 Factor principles and run using Platform. API first, use microservices to scale team

6 6



Pinned Tweet

Joshua McKenty @jmckenty · 30 Nov 2015

12 Factor: Separate code & config, stateful data & stateless processes, build steps from deployment; and trust operations to the platform.

21 20



Joshua McKenty @jmckenty · 30 Nov 2015

Continuous Delivery: Eliminate waste by letting customers use your work, push every commit to production. Only works for Cloud Native Apps.

2 4



Joshua McKenty @jmckenty · 30 Nov 2015

Toyoda's automatic shuttle changing loom was the first blue-green deploy system. Continuous Delivery since 1924.

1



Joshua McKenty @jmckenty · 30 Nov 2015

Containers: Better chroot.

6 13



Joshua McKenty @jmckenty · 30 Nov 2015

Cloud: Computing resources controlled by API.

8 8

HOW CAN
YOU
APPROACH IT?

TWO TYPES OF CLOUD NATIVE PLATFORMS

ASSEMBLED

UNSTRUCTURED

Ultra-Flexible

Multiple Configurations

Platform = Core Competency

Few Applications

DIY-focused

PRESCRIPTIVE

STRUCTURED

Standardized

Built-in Availability and Security

Automate & Simplify Operations

Many Applications

Buy-focused

TWO TYPES OF CLOUD NATIVE PLATFORMS

ASSEMBLED

UNSTRUCTURED



MESOS



kubernetes


PRESCRIPTIVE

STRUCTURED



CLOUD **FOUNDRY**

TWO TYPES OF CLOUD NATIVE PLATFORMS

ASSEMBLED UNSTRUCTURED		PRESCRIPTIVE STRUCTURED	
			
FEW	How many apps in the enterprise?	MANY	
Platform = Core Competency		AI	ns
YES	Is the platform a core competency?	NO	
kubernetes			

TWO TYPES OF CLOUD NATIVE PLATFORMS

ASSEMBLED

UNSTRUCTURED



PRESCRIPTIVE

STRUCTURED



PHILIPS



TWO TYPES OF INFRASTRUCTURE TO BUILD ON

BUILD IT YOURSELF

HYPER-CONVERGED

YES

Is racking, stacking, **and automating** a core competency?

NO



LET'S SEE HOW A
STRUCTURED PLATFORM
LOOKS LIKE

DEMO #2

THE STRUCTURED EXPERIENCE

[AKA cf push tc]

IN ESSENCE..
WHICH ARE THE LAYERS
IN THIS NEW WORLD?

LAYERS FOR CLOUD NATIVE APPLICATIONS

Cloud
Native
Platform

Structured or Unstructured

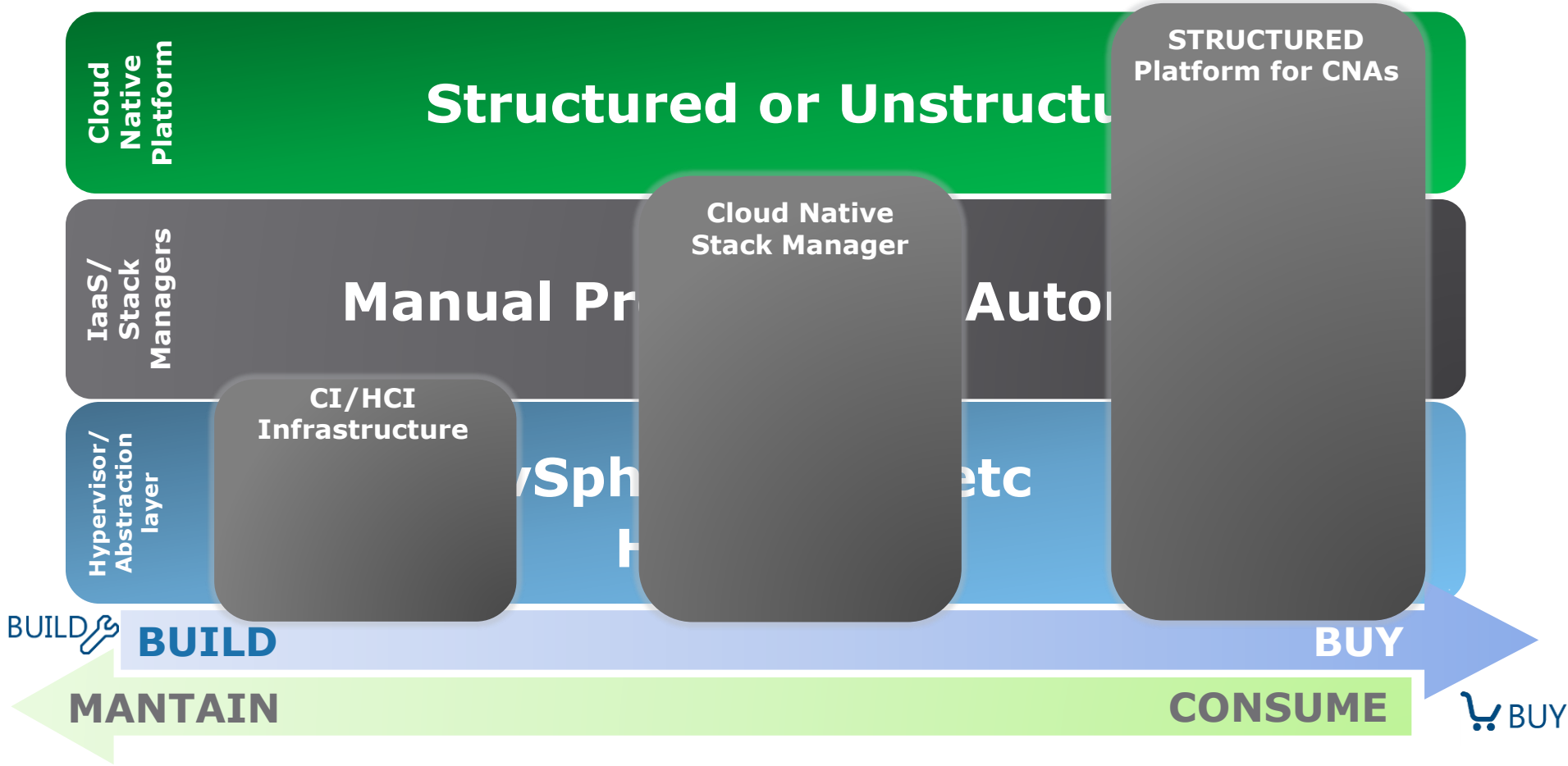
IaaS/
Stack
Managers

Manual Processes or Automation

Hypervisor/
Abstraction
layer

**vSphere/KVM/etc
Hardware**

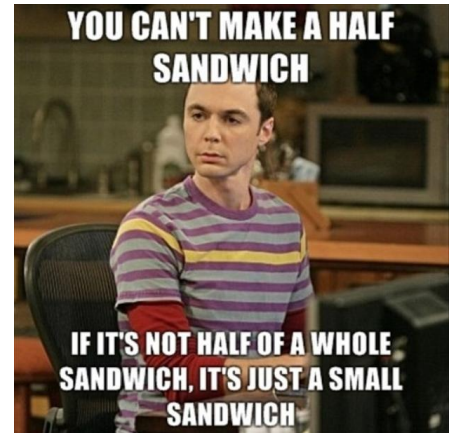
HOW TO SIMPLIFY CNAS DEPLOYMENT



*DIY= Do IT Yourself

RECAP / CLOSING COMMENTS

- ◇ Clear and Solid **Business Needs for Cloud Native Apps**
- ◇ Many Tech/options to choose from
- ◇ **Structured approach** offers many **advantages**
- ◇ **Purpose built** and **Simplified Infrastructure** can definitely help



THANK YOU!!!