

What is Cloud Native and why should I care?

Software Circus 2016



CLOUD NATIVE
COMPUTING
FOUNDATION

Open Source Cloud Computing *for Applications*

A complete and trusted tool kit for modern architectures

Who am I

Weaveworks CEO & CNCF TOC Chair

Previously:

Pivotal (Spring, vFabric, Tomcat, ...)

VMware (Redis, CloudFoundry, OpenStack)

RabbitMQ

Cohesive Networks

Metalogic

...



The excellent “Anywhere Exchange”



“Weave is a critical component in the International Securities Exchange (ISE) ‘Anywhere Exchange,’ providing software-defined networking for the multicast traffic within public cloud infrastructure,” said Rob Cornish, CIO of ISE. “Weave has turned a complex and demanding connectivity problem into ‘invisible infrastructure’ for our cloud-based disaster recovery solution. As a strategic vendor to ISE, we’re excited to see what Weave comes up with next.”

Lesson Learnt: *the tools have to be cloud native too*

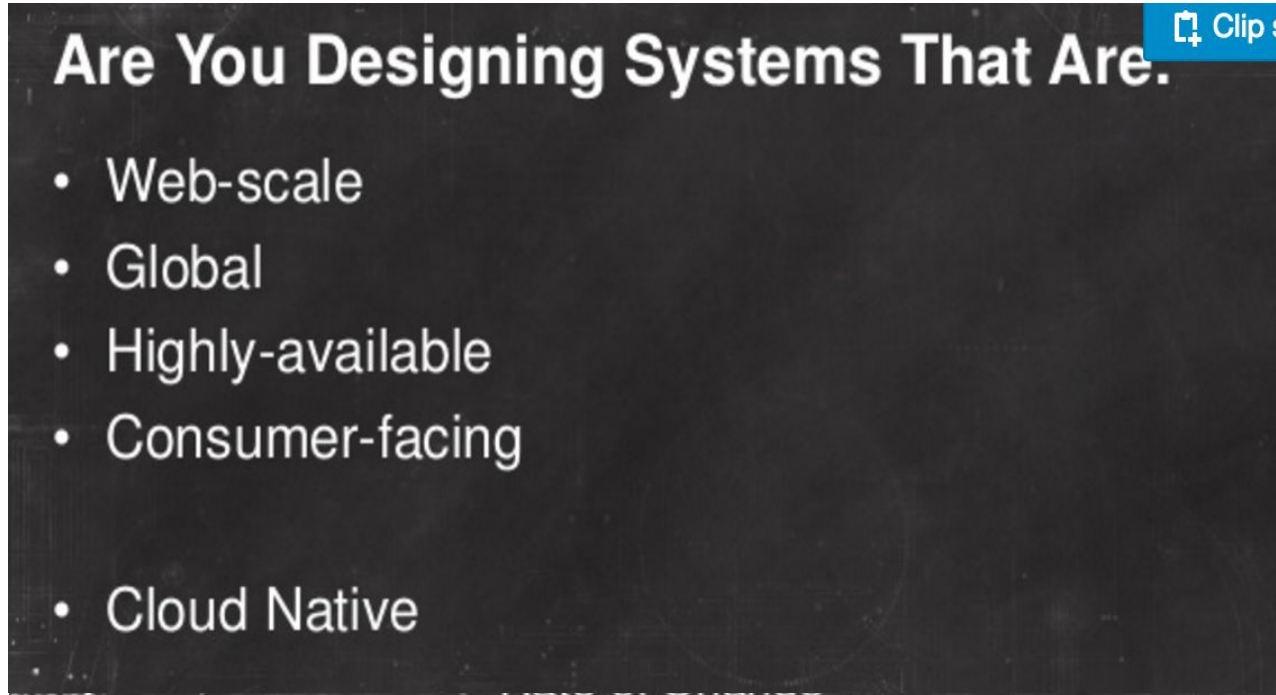
“Invisible Infrastructure”

Customers want to migrate applications to the cloud, but keep some parts behind the firewall, for flexibility or cost reasons

They want the ability to change cloud providers - no “lock in”

They want everything to “just work” so that they can focus on app

Netflix pioneered this



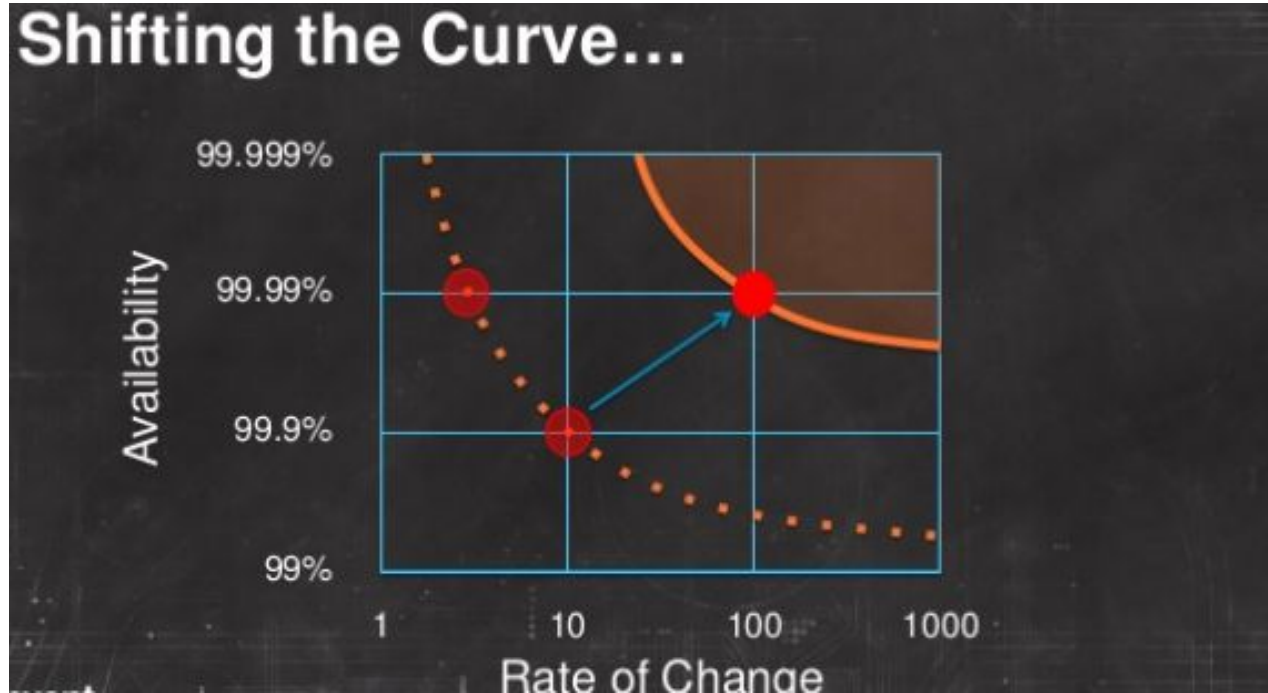
Are You Designing Systems That Are.

- Web-scale
- Global
- Highly-available
- Consumer-facing
- Cloud Native

The slide is a dark-themed presentation slide. It features a title at the top and a bulleted list of five characteristics. A small blue button with a white icon and the text 'Clip s' is located in the top right corner of the slide area.

<http://www.slideshare.net/AmazonWebServices/dmg206> for more - highly recommended

Netflix pioneered this



<http://www.slideshare.net/AmazonWebServices/dmg206> for more - highly recommended

Obligatory “Software is eating the world” pic..





The *Need for Speed*. This is real.

Figure 1

Comparison of IT performance metrics between high¹ and low performers

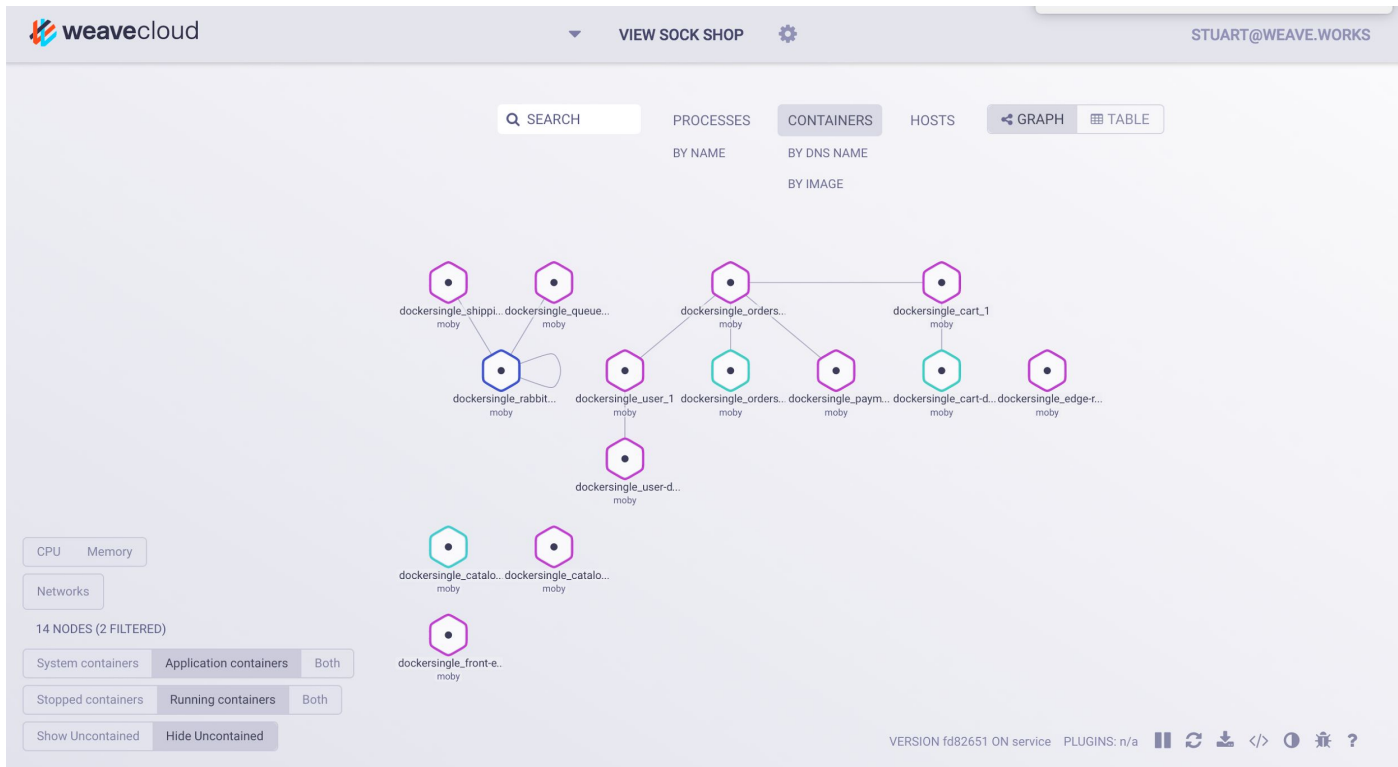
| | 2015 (Super High vs. Low) | 2014 (High vs. Low) |
|-----------------------------|---------------------------|---------------------|
| Deployment Frequency | 30x | 30x |
| Deployment Lead Time | 200x | 200x |
| Mean Time to Recover (MTTR) | 168x | 48x |
| Change Success Rate | 60x | 3x |

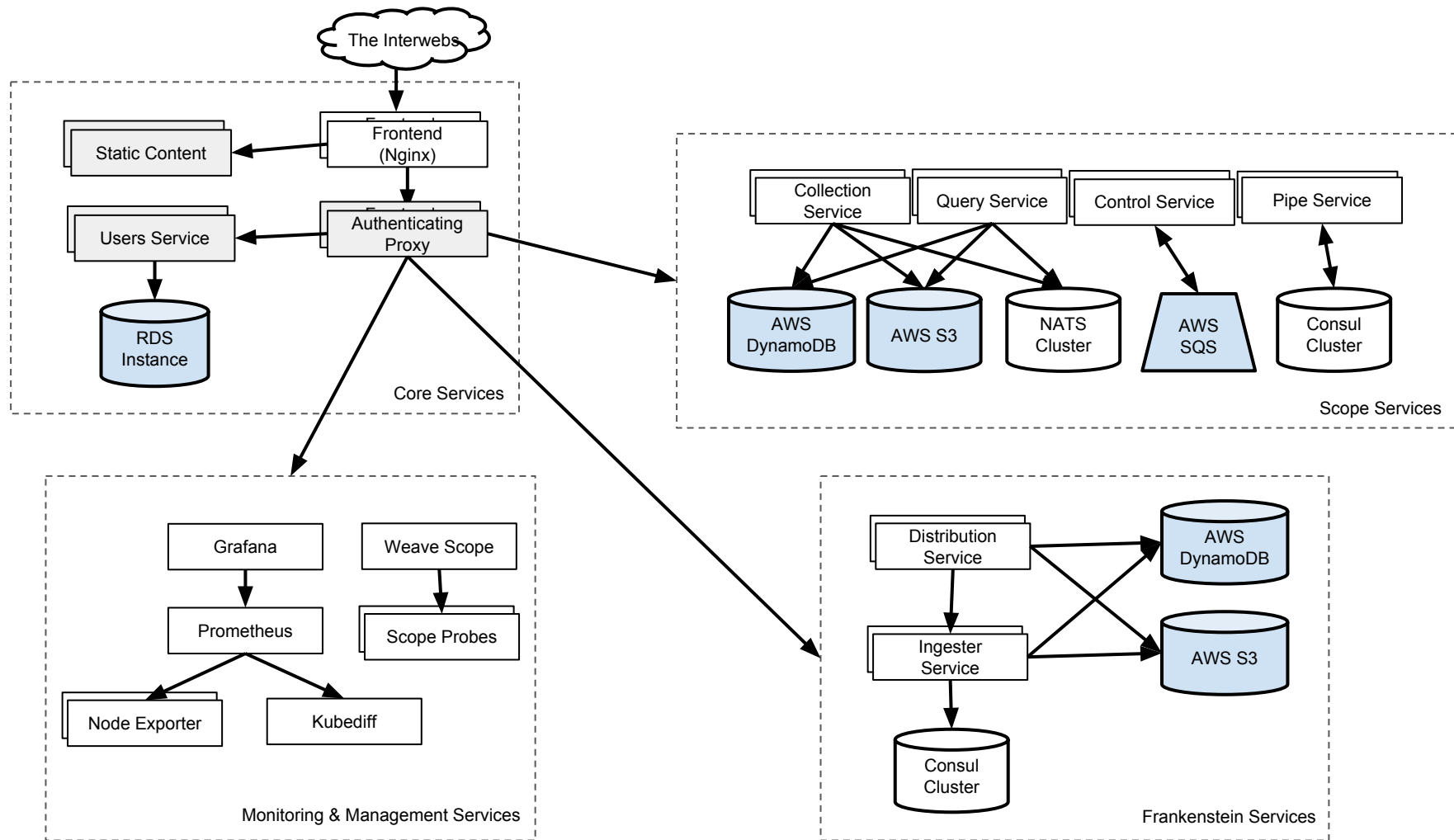
Puppet Labs state of devops 2015





Example: Weave Cloud - “Making Cloud Native Easy”





Key Points

24-7-365 Multitenant, Microservices, Automated, Secure.... ALL THE THINGS

We can focus on our application and not on Kubernetes / Docker / DCOS

We can run any component anywhere, not just on Amazon (*not quite yet!*)

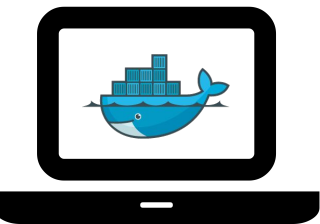
Independently scalable & composable: to help us keep cost low & be profitable

All the pieces work together e.g. Prometheus can monitor Docker, Kubernetes..

We happen to run on Kubernetes but can swap it out because pluggable etc.

Weave Cloud Development Process

App is developed
& tested locally



Built automatically
using CI of your
choice



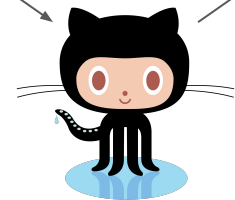
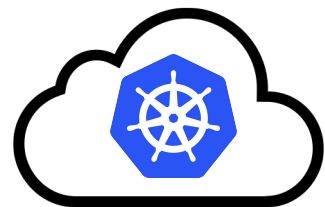
Container image
pushed
automatically



Deployed
automatically using
Weave Cloud deploy
service...



...to an
Environment
of your choice



Source control



Configuration as
code

Lesson Learnt: *Cloud Native needs good tools*

Open source

Run anywhere

Software you can trust, managed by credible teams & processes

Easy to monitor and control

Interoperates with other tools and common conventions

Lesson Learnt: *the infrastructure has to be boring*

To focus on your app, the infrastructure has to be boring.

Use PaaS/CaaS or any container platform you like.

Watch out for the 1% failure problem

Lesson Learnt: *We need good PATTERNS*

Microservices (and Microliths)

Cattle not Pets

Observability and Control baked in

Traffic Patterns - Blue/Green, Canary, smart routing & load balancing...

...

Cloud Native is **Patterns**



... and breathe ...

BUT BUT BUT

What is Cloud Native and
why should I care?



Open Source Cloud Computing *for Applications*

A complete and trusted tool kit for modern architectures

Cloud Native is **Patterns**

Patterns for what?

Availability

Automation

Acceleration

Anywhere!

Patterns for what?

Availability

Microservices & Netflix for everyone

Automation

Deployment & Management

Acceleration

CI/CD & OODA

Anywhere!

Containers are portable

Patterns need **Software**

Example: Management

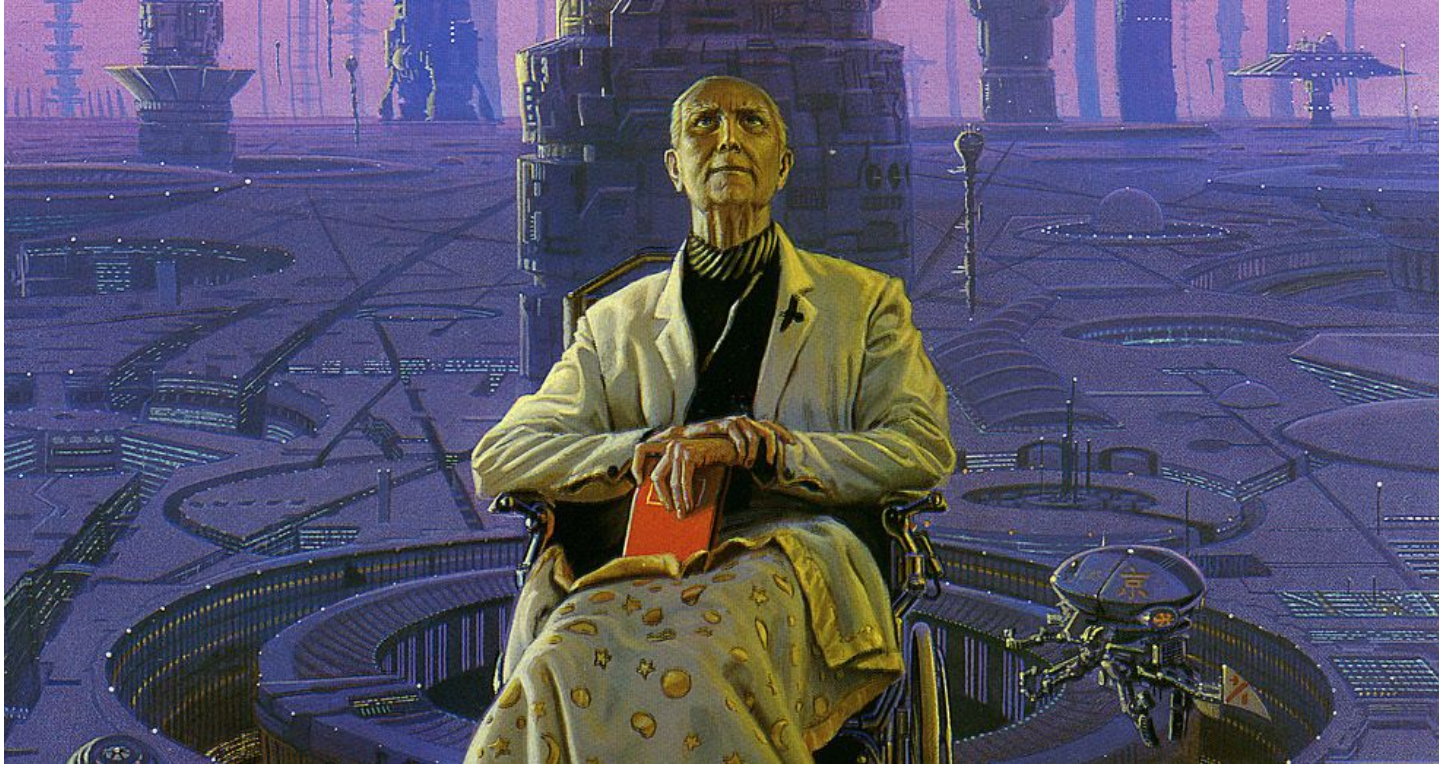


- Observability
 - View / Filter / Replay
 - Monitoring / Trace / Stream / Log
 - Business Intelligence
- Orchestration
- Coordination
 - Configuration
 - Discovery
 - DNS
- Service Management
 - Routing / Proxy / Load Balancer
 - Policy / Placement / Traffic Management

Assume we have the
software ... why do we need
a Foundation?

What even is a
Foundation?

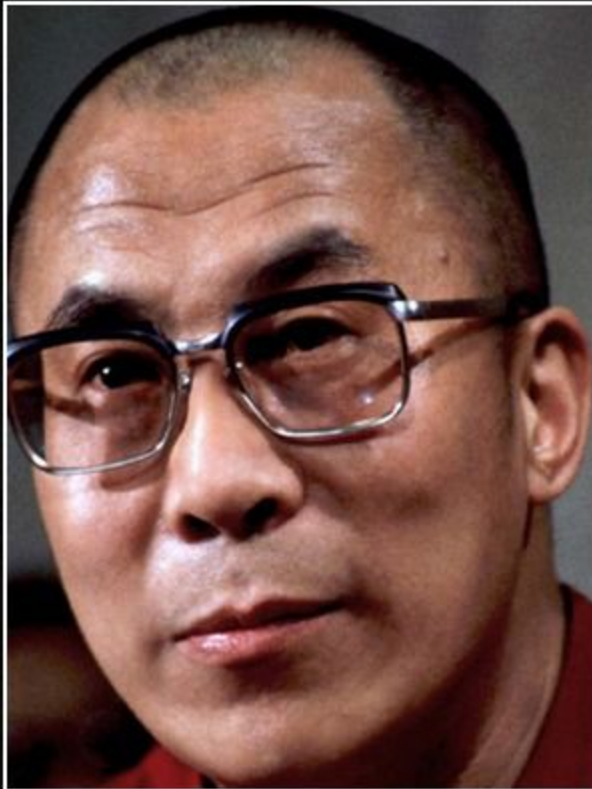
..a benevolent technocracy run by civilised robots?



..a federation of collaborating powers?



..a confluence of karmic forces?



You use force, you create fear. Fear destroys trust. Trust is the basis of harmony. The hardliner believes harmony and unity can be brought by force. That's totally unscientific, totally wrong.

— *Dalai Lama* —

AZ QUOTES

The Linux Foundation

Safeguards Linux for the long term

Provides a nexus for collaboration and trust

Is an ubiquitous open source brand

Good for customers & the community!

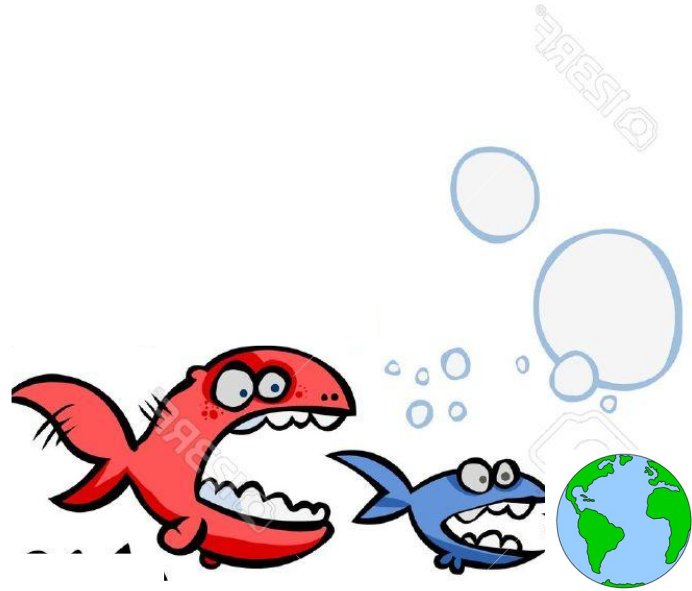
Common Open Source

is not proprietary

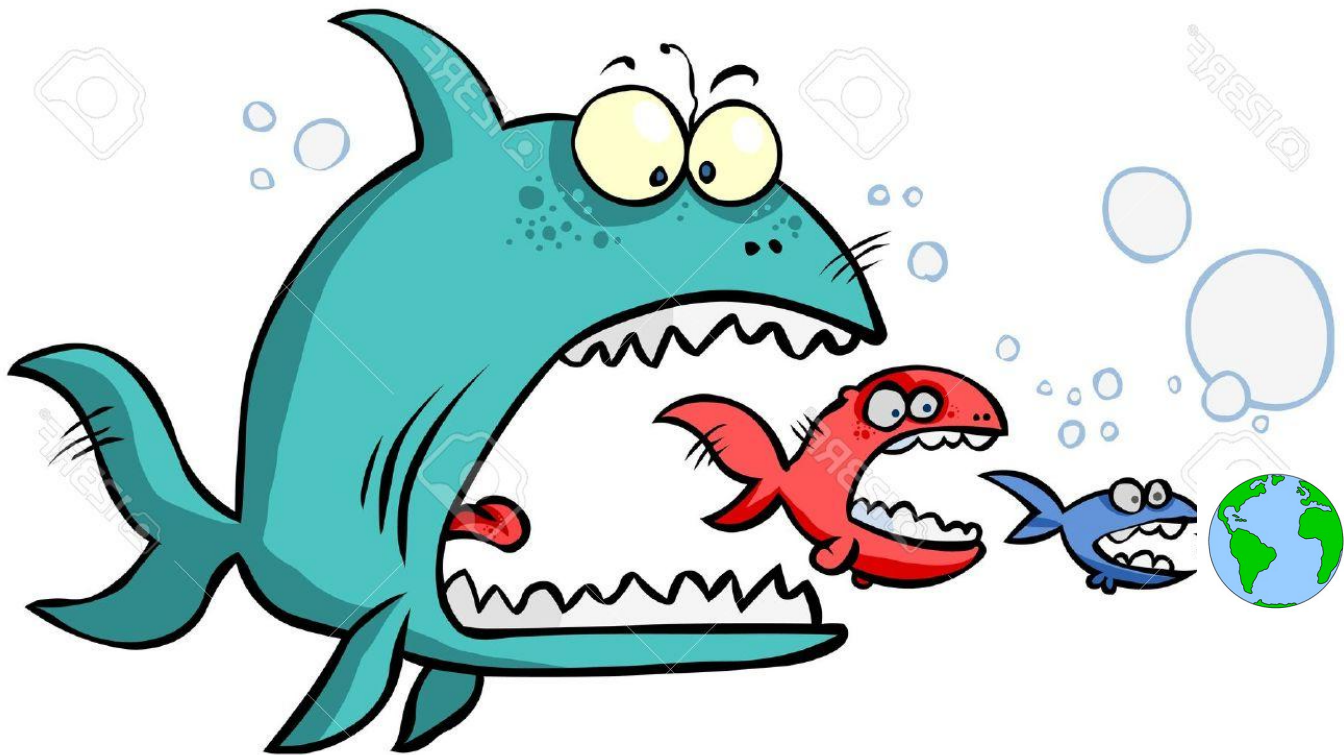
Software is eating the world



Open Source is eating Software



Cloud is eating Open Source



**Without a commons, we
risk Cloud Lock In**



Who does not want to be locked into one cloud

First it was the big software vendors

Then: big web companies like eBay, AirBnb

Now - big “traditional” companies too

→ Everyone wants to use open source, in a well managed commons, for the “boring infra” bits



CLOUD NATIVE
COMPUTING
FOUNDATION

Open Source Cloud Computing *for Applications*

A complete and trusted tool kit for modern architectures

CNCF for End Users

Easy - Fast - No Confusion - No Lock In

Guidance and clarity on “Cloud Native”

A badge of trust, quality & interoperability

A common set of tools, APIs & examples

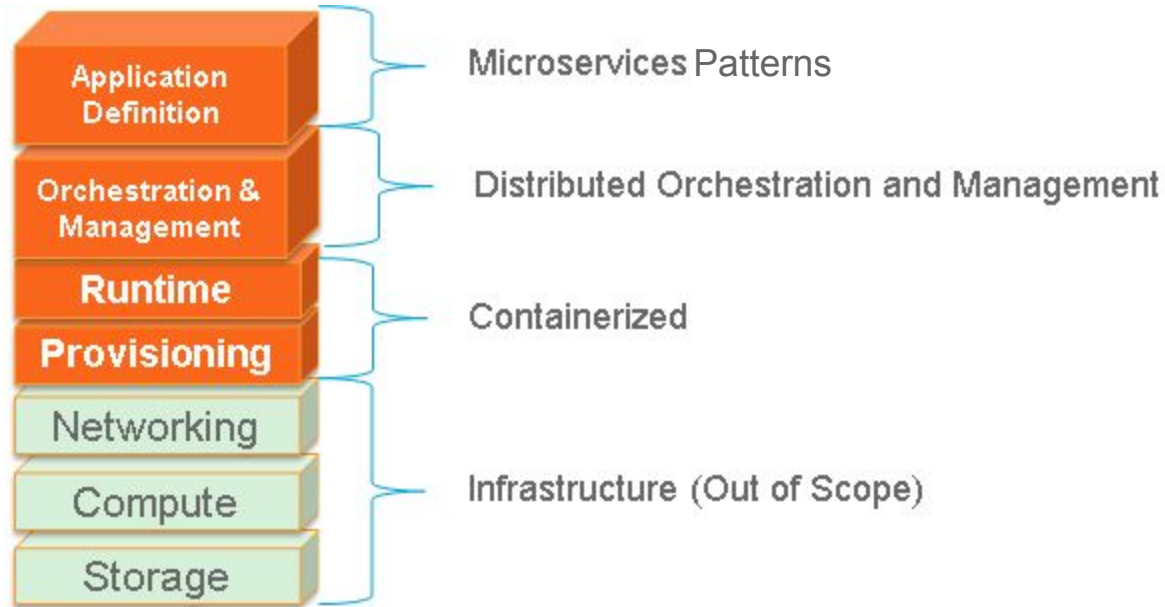
Shared through a modern, trusted commons

Tools you can trust

High quality, high velocity projects

- Kubernetes
- Prometheus
- Looking at: Fluentd, etcd, CoreDNS, OpenTracing, and more!

CNCF End User Reference Stack



Cloud Native Reference Architecture

Application Definition / Development

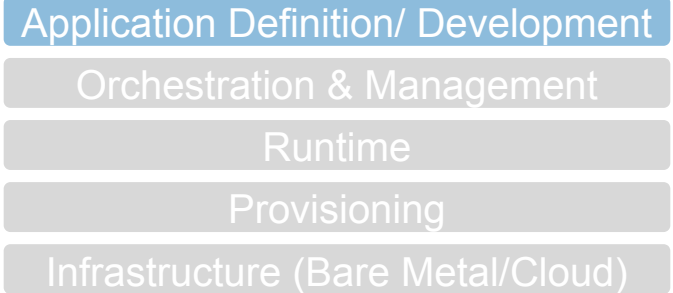
Orchestration & Management

Runtime

Provisioning

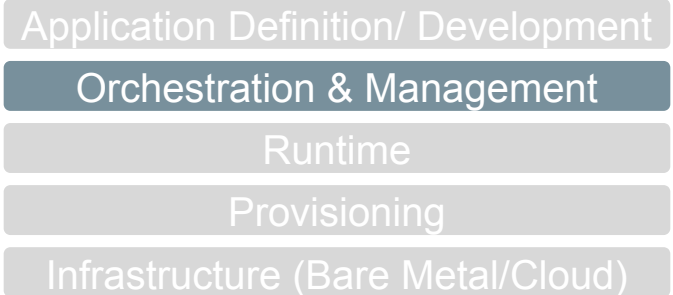
Infrastructure (Bare Metal/Cloud)

Application Definition/ Deployment Layer



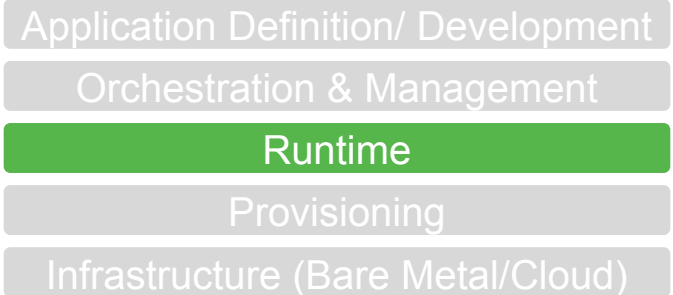
- Application Composition
- Application Delivery
- Application Development Frameworks
- Application Operational Tooling
- CI/CD
- Image Registry / Repository
- Governance and ops model

Orchestration & Management Layer



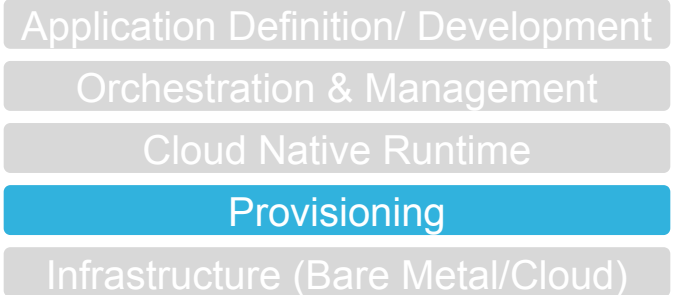
- Observability
 - View / Filter / Replay
 - Monitoring / Trace / Stream / Log
 - Business Intelligence
- Orchestration
- Coordination
 - Configuration
 - Discovery
 - DNS
- Service Management
 - Routing / Proxy / Load Balancer
 - Policy / Placement / Traffic Management

Runtime Layer



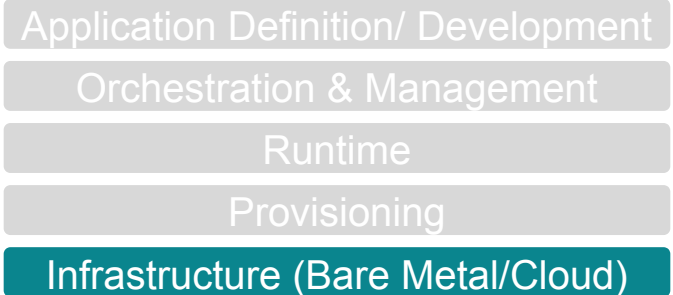
- Resource Management
 - Container Scheduling
 - Container Deployment
- Cloud Native – Network
 - Network Segmentation
 - SDN & APIs (eg CNI, libnetwork)
- Cloud Native – Data
 - Data Management
 - Databases & APIs
- Overall Container Service
 - (Some) PaaS/Platform Services

Provisioning Layer



- OS Management
- Secure Images
- Host level Devops Deployment Tooling & Provisioning

Infrastructure (Bare Metal/Cloud) Layer



- Out of scope for CNCF projects as we do not define infrastructure vendors or cloud solutions but part of reference architecture
- Potentially in the future we will provide “certification”

The **S** word

Standards

Standards are...

An algorithm for identifying areas of disagreement and maximising them.

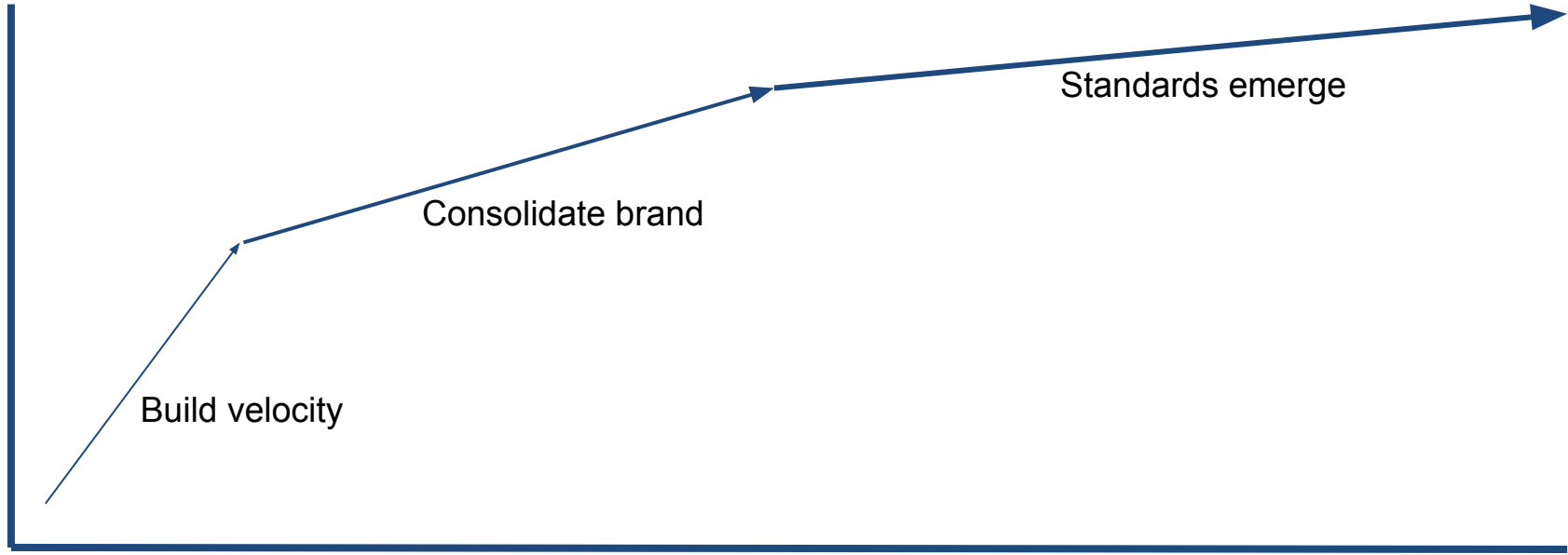
Standards are slow. And emerge slowly.

CNCF does not need standards, we need ease of interop and “glue” code

We like conventions and normal forms, and arise from real world use by the community - like CNI for example.

We can document these as specs and others, if they wish, may create IETF RFCs for example.

3 phase plan



Docker

Bob Wise, Samsung: “An Ode To Boring”



“I call on the CNCF to formally foster a common community container implementation project backed by the Kubernetes, Mesos, and Cloud Foundry communities.

We need a transparent, community-driven implementation ... to become the default container implementation for a wide number of open source orchestration systems”

https://medium.com/@bob_48171/an-ode-to-boring-creating-open-and-stable-container-world-4a7a39971443#.2w2edyeir



What the ecosystem is asking for

A core standard container (appc)

Stability - LTS releases?

An open platform

Docker is important!

Application Definition / Development

Docker Platform

Docker Runtime

Provisioning

Infrastructure (Bare Metal/Cloud)

Parting thought: Do you want Container Lock in?

Docker is portable across linux, windows, amazon and vmware

This is awesome!

But it opens up space for “one platform to rule them all”

Do we want competition at this layer? Or common plumbing? Or both?



Open Source Cloud Computing *for Applications*

A complete and trusted tool kit for modern architectures