Microservices in the Cloud using Kubernetes, Docker and Jenkins

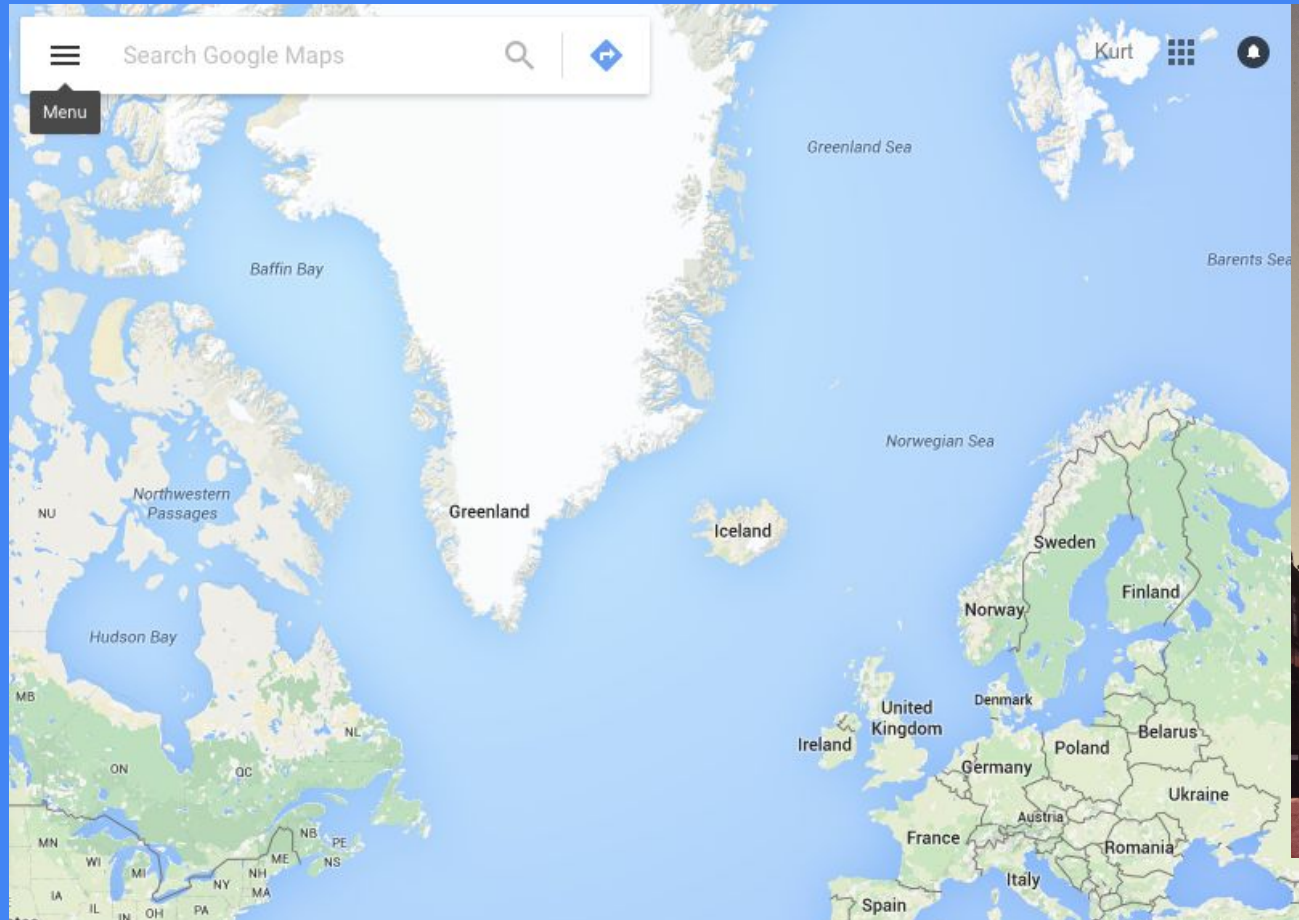
SATURN May 3rd, 2017

@KurtStam, PhD,
Principal Engineer on the #Fabric8/Fuse team

Content

- MicroAdventures & MicroServices
- Introduction to Docker
- Introduction to Kubernetes/OpenShift
- Demo of RPi Cluster running K8s
- Jenkins: Fabric8 CI/CD Pipeline

Spitsbergen Exp 1985




SPITSBERGEN '85

redhat.

redhat.

#microadventures

#microadventures

#microadventures

Monolith and Conventional Deployment

Lots of planning
Different teams and responsibilities
Regression Issues
Cheap to start, hard to maintain when you hit a certain complexity level

Micro Services

- Micro Services are about time to market
- Component reuse, not code reuse.
- ‘One concern’: Simple and small, but not too small
- Easy to test, limited risk of regressions, CI/CD
- One team from development to deployment
- API Contract (REST & Swagger), API Manager
- Perfect for cloud deployment!

Java Micro Services

- AngularJS UI for display logic
- REST Service(s) Swagger 2 for business backend
- SQL/No-SQL store & Caching

Open Source Cloud: Virtualization of the entire stack

- Fabric8: iPaas
- OpenShift: Paas
- Kubernetes: Docker Orchestration
- Docker OS Level Virtualization

Open Source Cloud: Virtualization of the entire stack

Shipping Containers At Clyde, by Steve Gibson

docker

Shipping software is hard

Multiplicity of Stacks

Static website

nginx 1.5 + modsecurity + openssl + bootstrap 2

User DB

postgresql + pgv8 + v8

Queue

Redis + redis-sentinel

Analytics DB

hadoop + hive + thrift + OpenJDK

Background workers

Python 3.0 + celery + pyredis + libcurl + ffmpeg + libopencv + nodejs + phantomjs

Web frontend

Ruby + Rails + sass + Unicorn

API endpoint

Python 2.7 + Flask + pyredis + celery + psycopg + postgresql-client

Do services and apps interact appropriately?

Multiplicity of hardware environments

Development VM

QA server

Customer Data Center

Public Cloud

Disaster recovery

Production Cluster

Production Servers

Contributor's laptop

Can I migrate smoothly and quickly?

Matrix from Hell

	Development VM	QA Server	Single Prod Server	Onsite Cluster	Public Cloud	Contributor's laptop	Customer Servers
Static website	?	?	?	?	?	?	?
Web frontend	?	?	?	?	?	?	?
Background workers	?	?	?	?	?	?	?
User DB	?	?	?	?	?	?	?
Analytics DB	?	?	?	?	?	?	?
Queue	?	?	?	?	?	?	?

Analogy with Cargo Transport Pre-1960

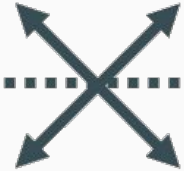
Multiplicity of Goods







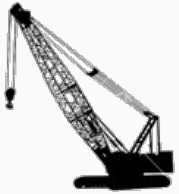
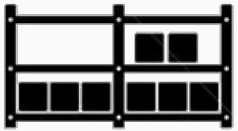
Do I worry about how goods interact (e.g. coffee beans next to spices)



Multipilicity of methods for transporting/storing





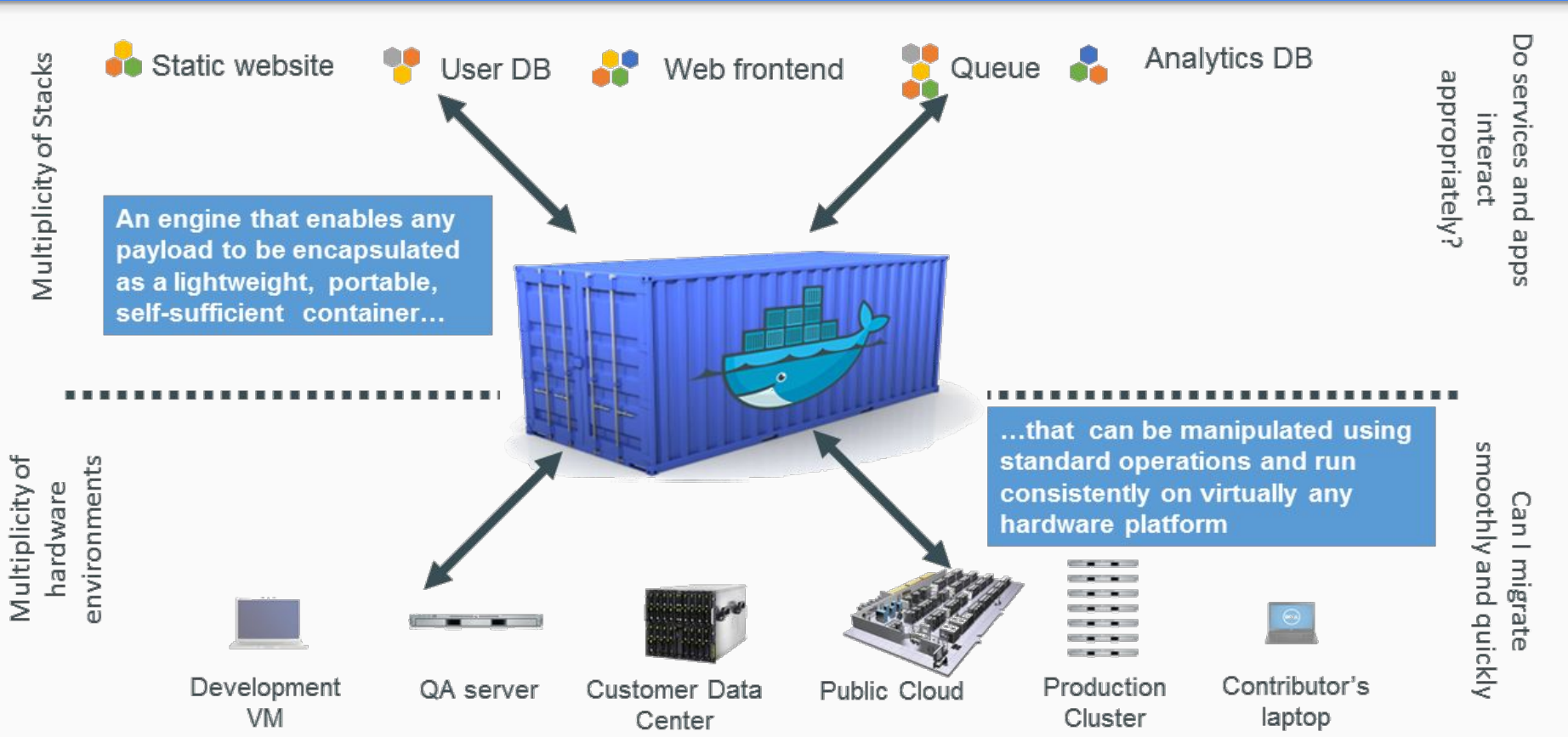

Can I transport quickly and smoothly (e.g. from boat to train to truck)

Same Matrix from Hell

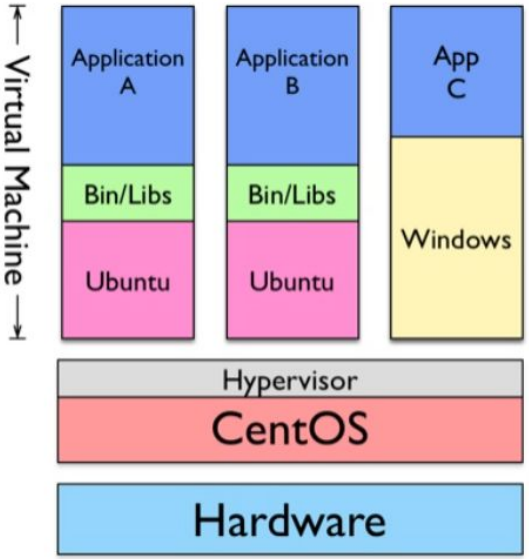
	?	?	?	?	?	?	?
	?	?	?	?	?	?	?
	?	?	?	?	?	?	?
	?	?	?	?	?	?	?
	?	?	?	?	?	?	?
	?	?	?	?	?	?	?
							

Solution: Intermodal Shipping Container

Docker is the Shipping Container for Code

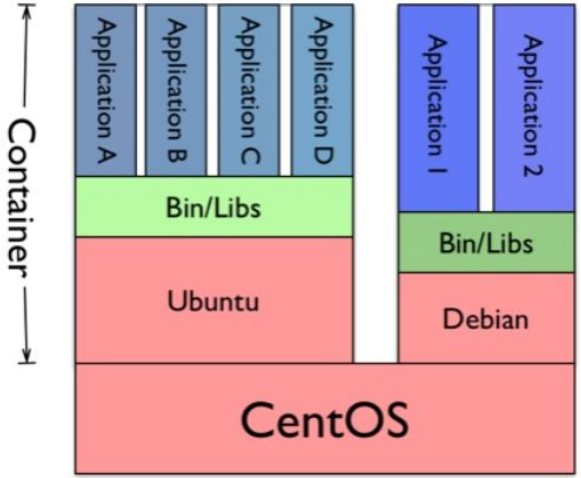


Lightweight Container vs. VM



Containers are isolated, but sharing the kernel and (some) files

➜ faster & lighter



Docker Demo: Hello World

Base Image + Shared Kernel
Process Isolation
Layers: pull, commit, push
DockerHub

```
Centos: yum install docker

docker run centos echo hello world

docker run -it centos bash

https://hub.docker.com/r/kurtstam/saturn
```

Docker Demo: Dockerfile


```
FROM php:5.6-apache                    (https://hub.docker.com/_/php/)
COPY src/ /var/www/html/
```

```
docker build -t php-hello-world .
```

```
docker run -it -p 80:8001 php-hello-world
```

Computational Resources: Cloud

Kubernetes

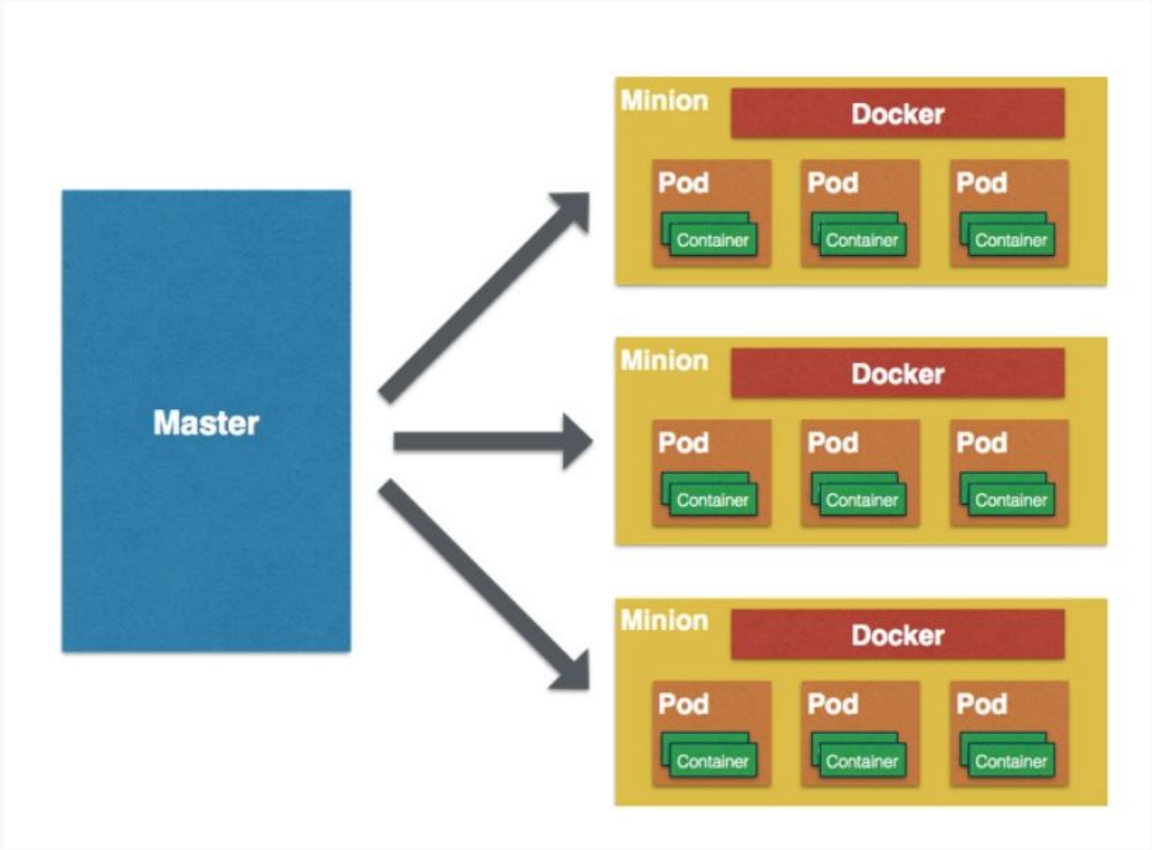
Kubernetes: ‘Helmsman of a ship’
based on Borg experiences

Container (Docker, Rocket) Orchestration
Cloud Operating System

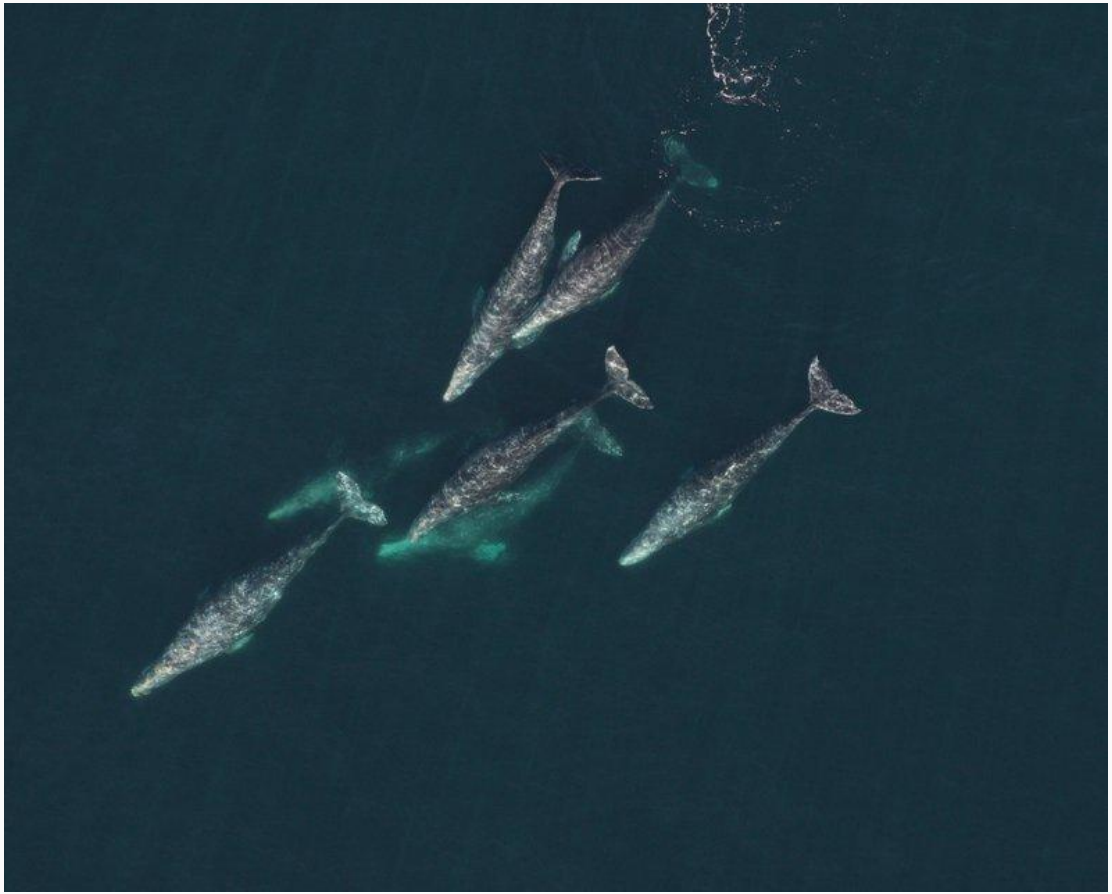
Three flavors:

- OpenShift OnLine (Public PaaS) running on Amazon, Google, etc clouds
- OpenShift Enterprise (Private Paas), running in your data center
- Origin (Community Paas), running on a laptop (MiniKube, MiniShift)

Kubernetes


Master
Minion
Docker
Pod
Container
Pod
Container
Pod
Container
Minion
Docker
Pod
Container
Pod
Container
Pod
Container
Minion
Docker
Pod
Container
Pod
Container
Pod
Container

Kubernetes Pod



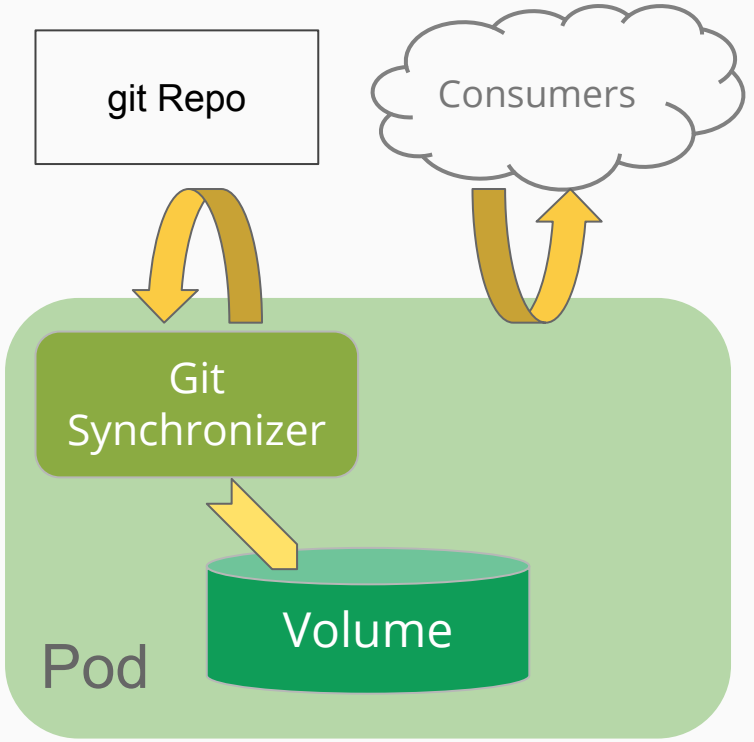
Kubernetes Pod

A Pod contains one or more containers

Containers within a pod are **tightly** coupled

Shared namespaces

- Containers in a pod share IP, port and IPC namespaces
- Containers in a pod talk to each other through localhost



Kubernetes Pod Networking

Pods have IPs which are routable

Pods can reach each other without NAT

- Even across nodes

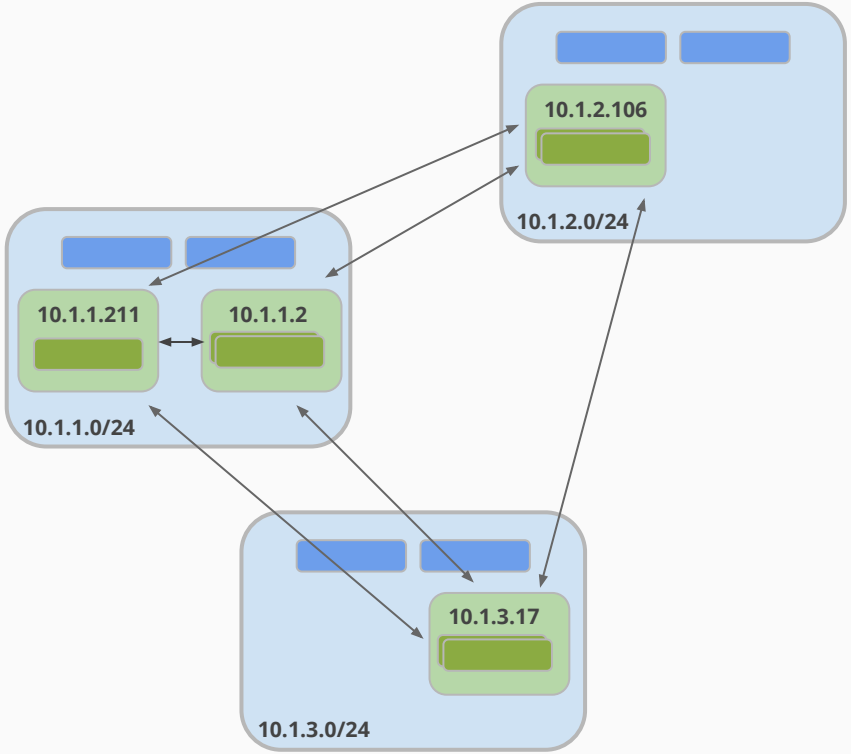
No Brokering of **Port Numbers**

These are fundamental requirements

Many solutions

- Flannel, Weave, OpenVSwitch, Cloud Provider

Let's deploy a pod!



Kubernetes Service

A logical grouping of pods that perform the same function

- grouped by label selector

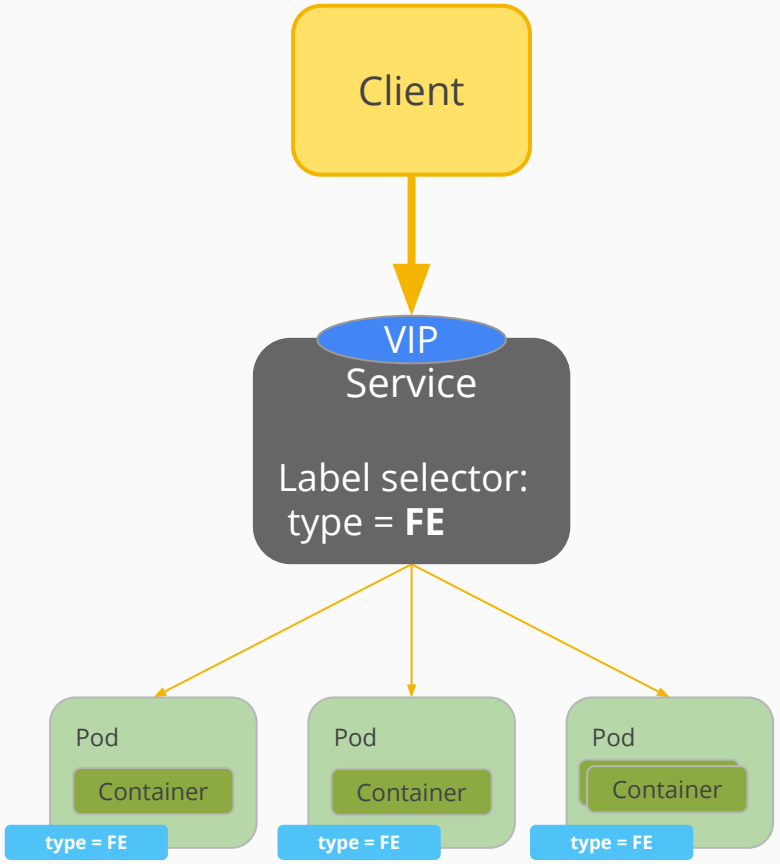
Load balances incoming requests across constituent pods

Choice of pod is random but supports session affinity (ClientIP)

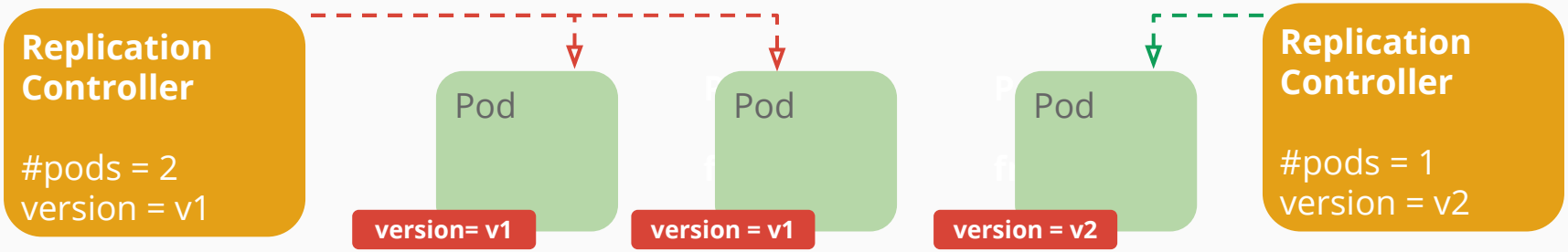
Gets a **stable** virtual IP and port

- also a DNS name

Let's deploy a service!



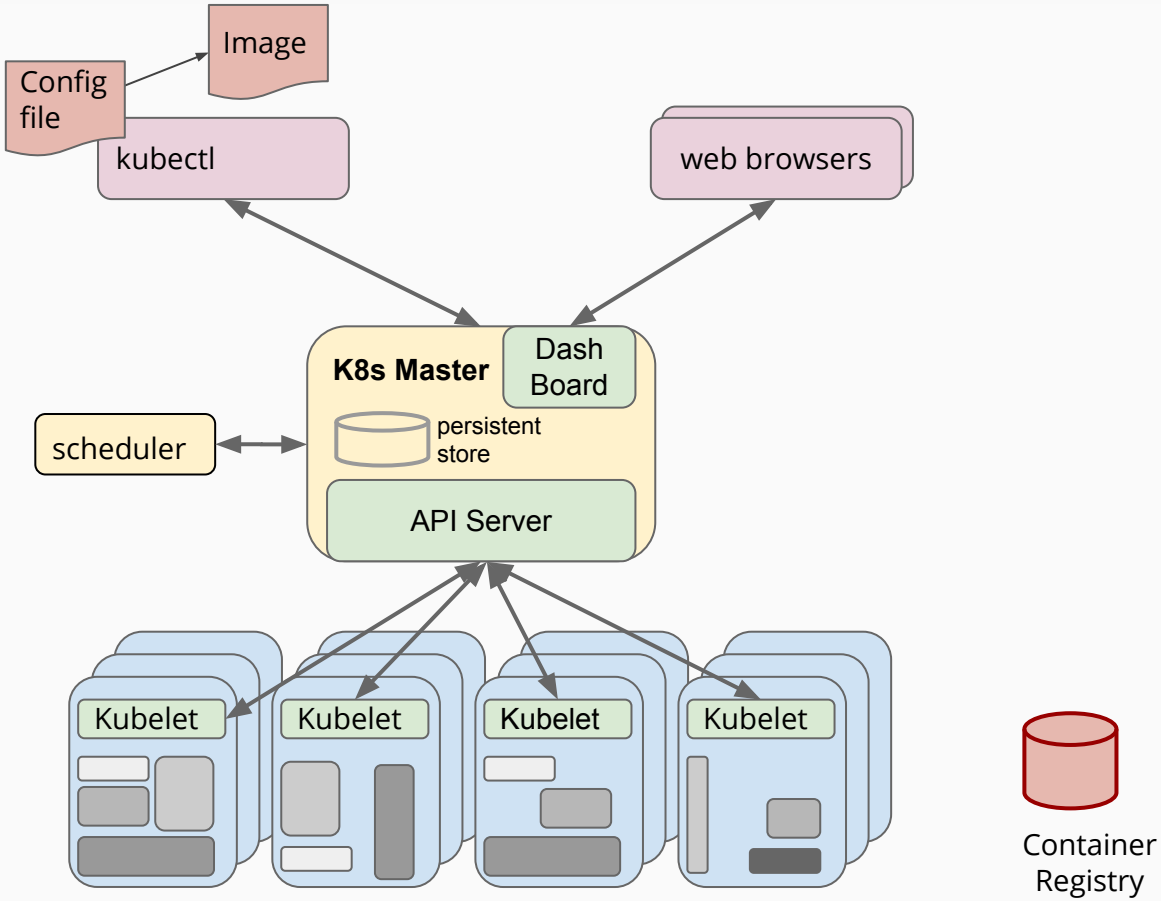
Kubernetes ReplicationController



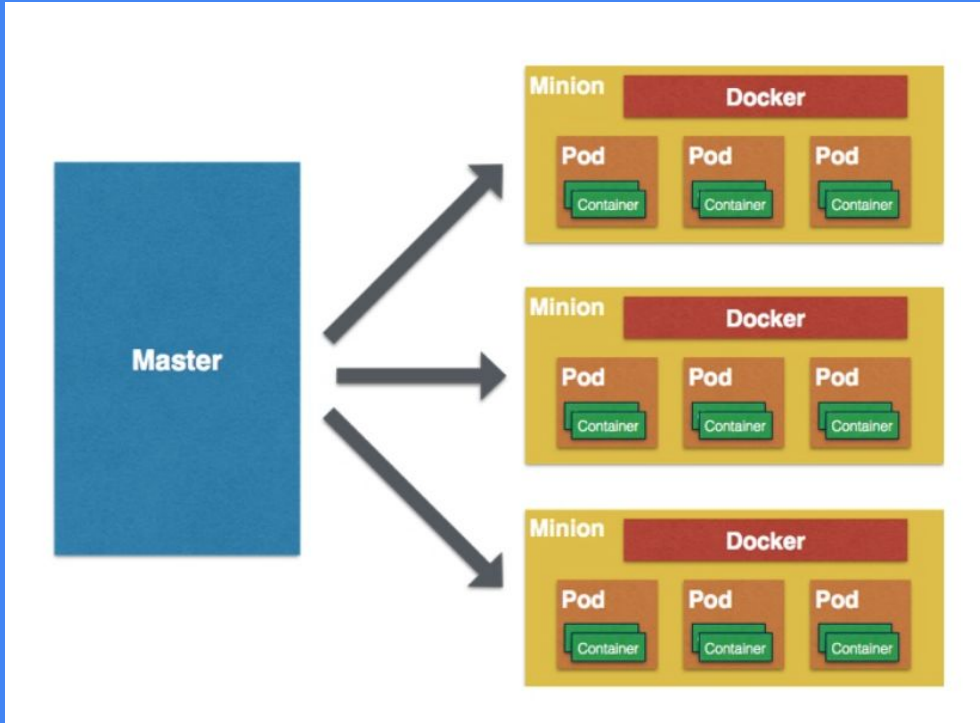
- Keeps Pods running
- Gives direct control of Pod #s
- Grouped by Label Selector

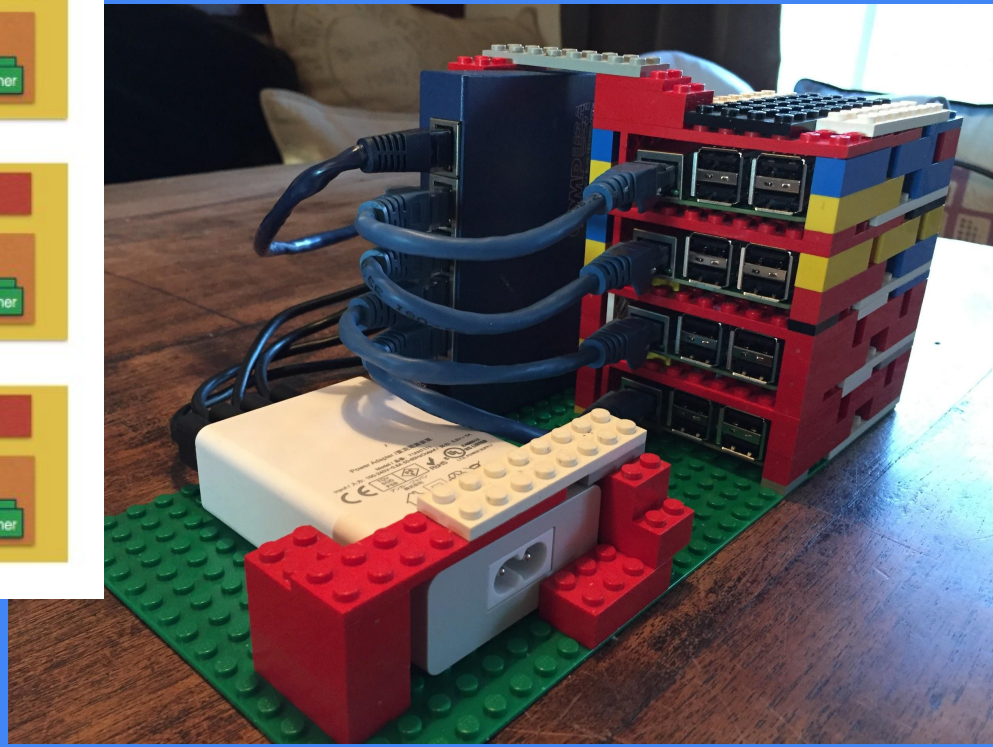
Let’s scale a service!

Developer View



Kubernetes on RaspberryPi





http://www.github.com/Project31

Shopping List

To build this four-Pi setup I used:

- 4 Raspberry Pi 2s
- 4 16GB MicroSD cards (Class 10)
- 1 60W power supply with USB outlets
- 4 short USB to Micro USB cables (for powering the Pis)
- 4 short Cat 5 network cables
- 1 longer Cat 5 network cable to hook into your network
- 1 network hub (Mine is an old five-port, 10/100MBps I dusted off)
- LEGOs (Trust me, it feels good to build your own!)

Fabric8

Fabric8 Management:

Hawtio Console, Logging, Metrics, Maven plugin

---------- containers ----------

Fabric8 iPaas:

ActiveMQ Messaging, Camel, **API management**

iPaaS Quickstarts:

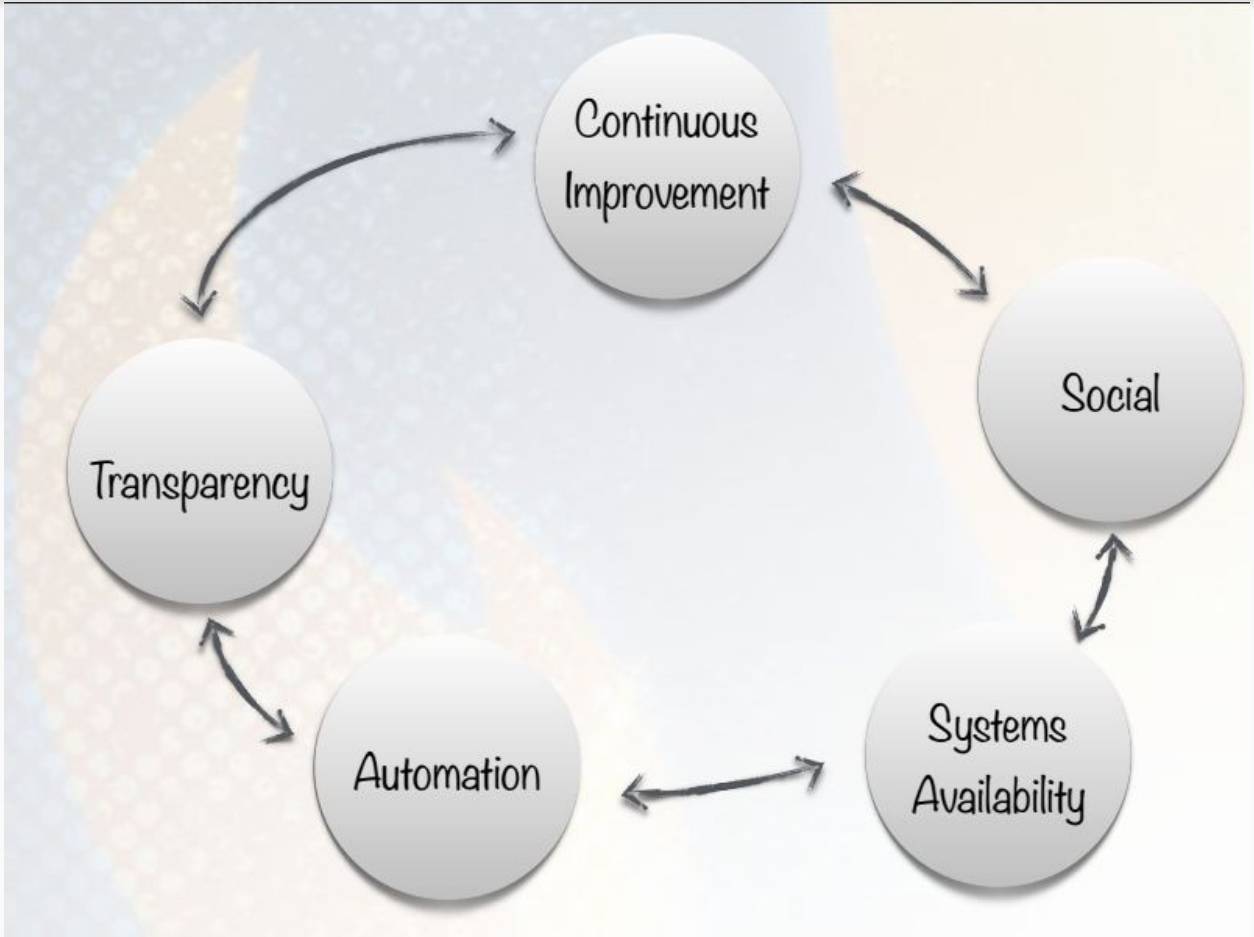
micro service examples

Fabric8 DevOps:

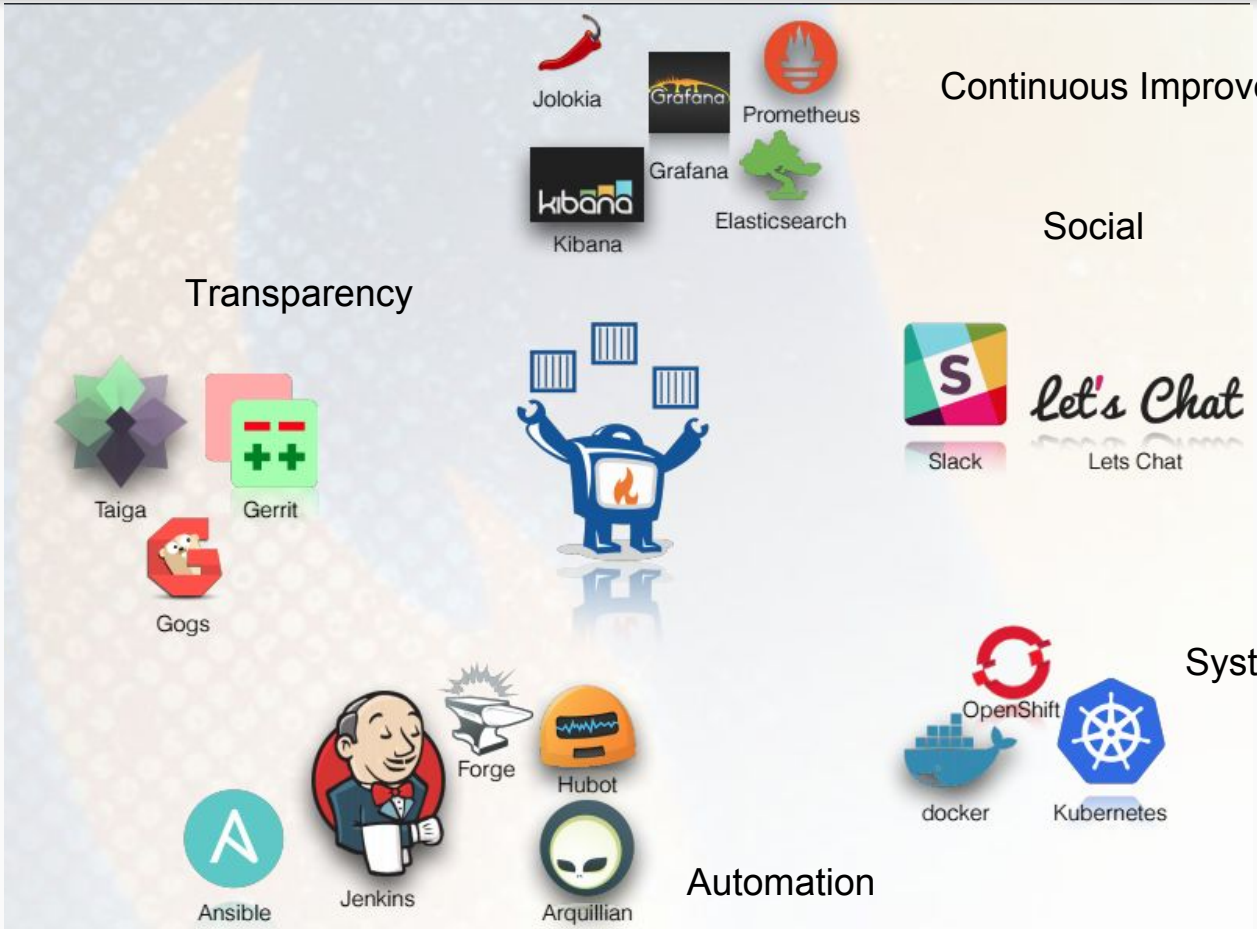
CD pipeline, jenkins, gogs, chat, gerrit, hubot

(q)

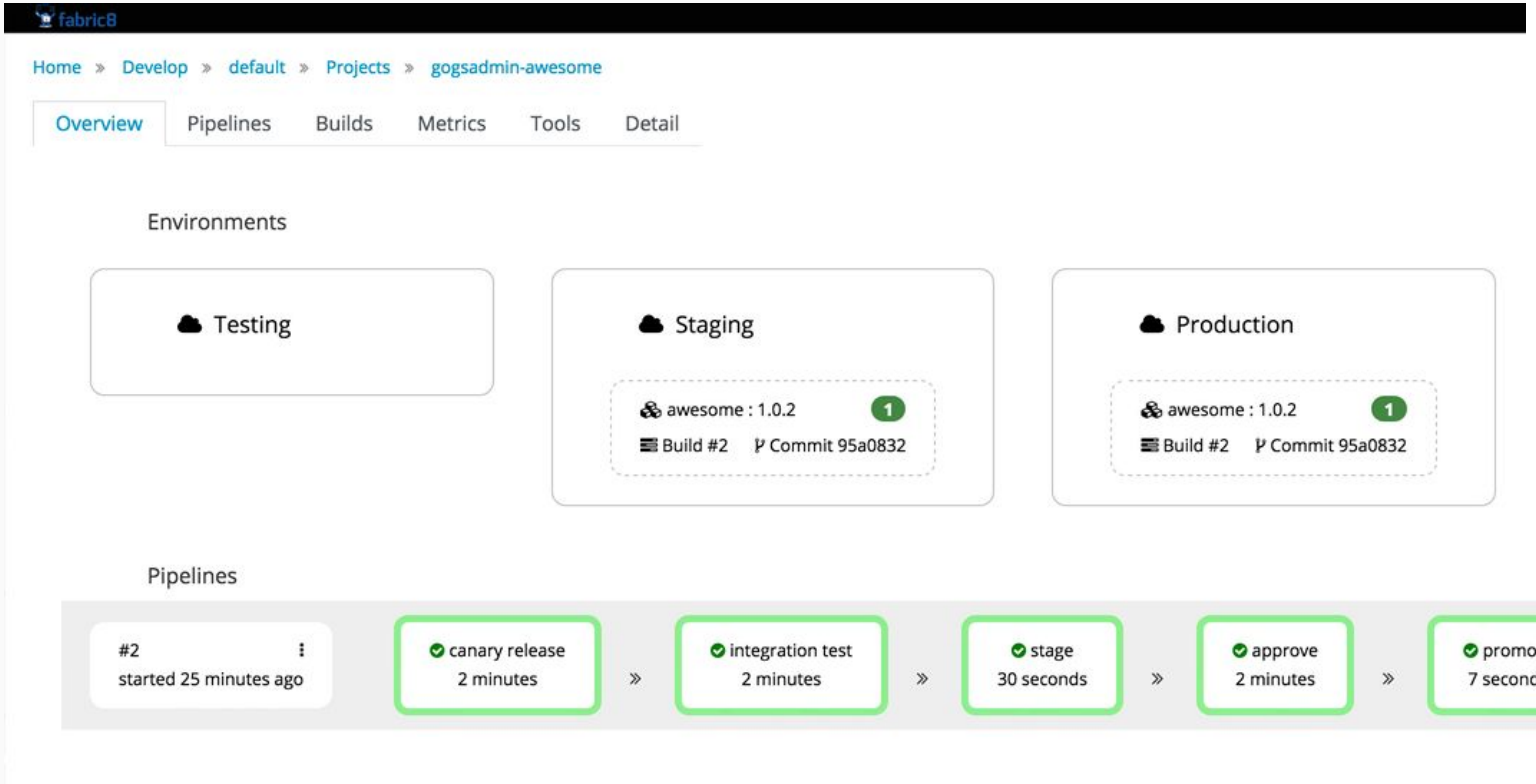
Developer Experience



Developer Systems



Continuous Delivery Pipeline using Fabric8 and Jenkins Demo



Questions?

Fabric8.io	Fabric8 Microservices Platform
Get Started	Create a Kubernetes Cluster with Fabric8
Docker.com	Create and Run Container Images
Kubernetes.io	Container based Cloud
Openshift.org	Red Hat product based on Kubernetes

@KurtStam

References

[1] http://www.fabric8.io

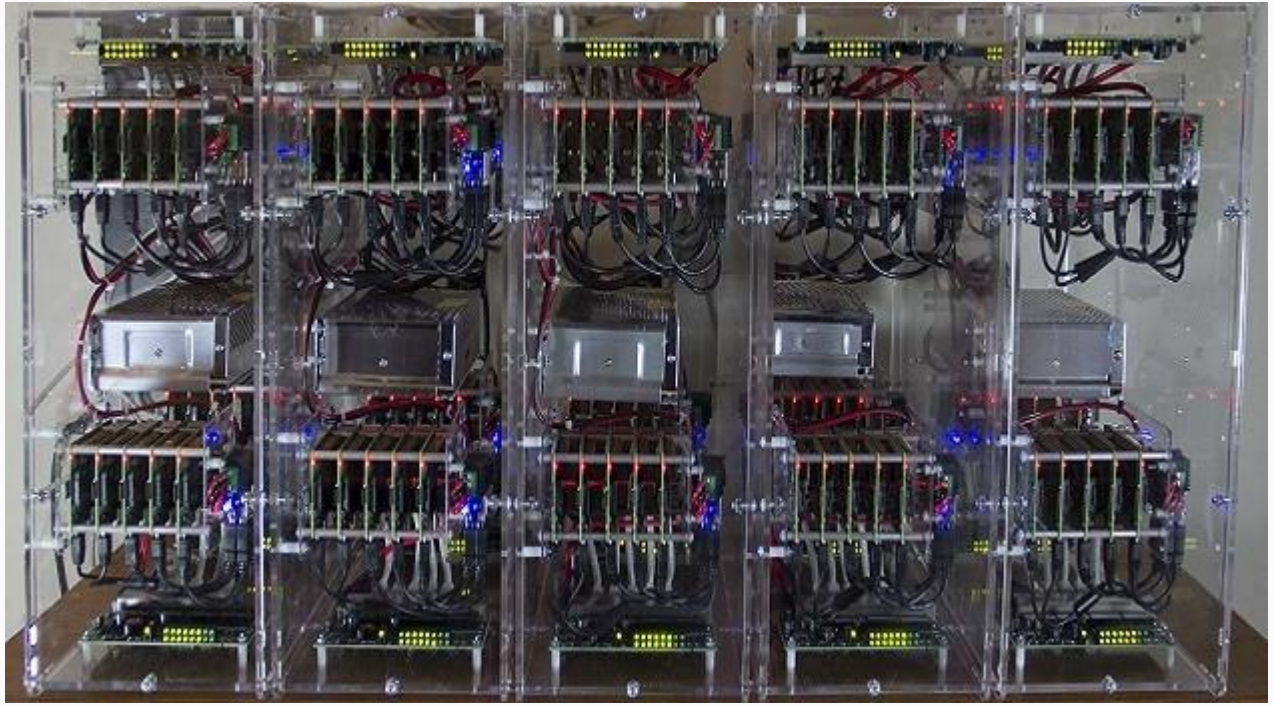
[2] http://www.apiman.io

[3] @tekggrl "Kubernetes: From Beginner to Expert"

[4] http://www.github.com/Project31

[5] https://opensource.com/life/16/2/build-a-kubernetes-cloud-with-raspberry-pi

Pico Cluster



100 RPi boards:

400 Cores
400 GB Ram

Storage on clustered MicroSD or SAN

8,000 $

Disruptive technology: Platform of tomorrow

Open Source Everything
Low cost
Low power
Redundant
Distributed
Super scalable