Microservices in the Cloud using Kubernetes, Docker and Jenkins

SATURN May 3rd, 2017

@KurtStam, PhD,
Principal Engineer on the #Fabric8/Fuse team
MicroAdventures & MicroServices
Introduction to Docker
Introduction to Kubernetes/OpenShift
Demo of RPi Cluster running K8s
Jenkins: Fabric8 CI/CD Pipeline
#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 software is hard
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Development VM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QA Server</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Single Prod Server</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Onsite Cluster</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Public Cloud</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Contributor's laptop</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Customer Servers</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Analogy with Cargo Transport Pre-1960

Multiplicity of Goods

- Wooden crate
- Electronic devices in a container
- Chemical drums
- Piano

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

Multiplicity of methods for transporting/storing

- Truck
- Forklift
- Train
- Crane
- Ship

Can I transport quickly and smoothly (e.g., from boat to train to truck)?
<table>
<thead>
<tr>
<th><img src="image" alt="Barrels" /></th>
<th><img src="image" alt="Crates" /></th>
<th><img src="image" alt="Cars" /></th>
<th><img src="image" alt="Barrels" /></th>
<th><img src="image" alt="Crates" /></th>
<th><img src="image" alt="Cars" /></th>
<th><img src="image" alt="Barrels" /></th>
<th><img src="image" alt="Crates" /></th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Train" /></td>
<td><img src="image" alt="Store" /></td>
<td><img src="image" alt="Loader" /></td>
<td><img src="image" alt="Loader" /></td>
<td><img src="image" alt="Ship" /></td>
<td><img src="image" alt="Ferry" /></td>
<td><img src="image" alt="Ferry" /></td>
<td><img src="image" alt="Truck" /></td>
</tr>
</tbody>
</table>

**Same Matrix from Hell**
Solution: Intermodal Shipping Container

A standard container that is loaded with virtually any goods, and stays sealed until it reaches final delivery.

...in between, can be loaded and unloaded, stacked, transported efficiently over long distances, and transferred from one mode of transport to another.

Multiplicity of Methods for Transporting/Storing

Do not worry about how goods interact (e.g., coffee beans next to spices).

Can transport quickly and smoothly (e.g., from boat to train to truck).
Docker is the Shipping Container for Code

An engine that enables any payload to be encapsulated as a lightweight, portable, self-sufficient container...

...that can be manipulated using standard operations and run consistently on virtually any hardware platform.
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
COPY src/ /var/www/html/
docker build -t php-hello-world .
docker run -it -p 80:8001 php-hello-world
Computational Resources: Cloud
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
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
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!
Kubernetes on RaspberryPi

http://www.github.com/Project31
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)
- LEGOos (Trust me, it feels good to build your own!)
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
Developer Experience

- Transparency
- Automation
- Continuous Improvement
- Social
- Systems Availability
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


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


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