Managed Containers, Open Source, and Google

2014 CloudOpen/LinuxCon

Joe Beda
Senior Staff Software Engineer

jbeda@google.com
@jbeda
Why Containers*?

1. Packaging
2. Efficiency and Speed
3. Security (?)

(*) Container = Docker flavor container
Packaging

Static application environment
No stress deployment and update

Repeateable portable artifact
Develop here, run there
Pick your cloud solely on its merits

Loosely coupled
= easier to build and manage
Easier to build and manage
Compose applications from micro-services
Everything at Google runs in a container.

- Resource isolation
- Predictability
- Quality of service
- Efficient overcommit
- Resource accounting

Google starts over 2 billion containers per week.
Kubernetes

κυβερνήτης: Greek for “pilot” or “helmsman of a ship”
the open source cluster manager from Google
Kubernetes

*Inspired by Google's systems and experience*

Manage Containers, not Machines

**Efficient**: optimized packing, better scaling

**Performant**: active environment tuning

**Robust**: active monitoring, self healing

**Organizationally Scalable**:
- Split sys ops and app ops
- Enable micro-services

**Modern Open Source**:
- Extensible & portable, can run anywhere
- Apache 2.0 licensed
- Written in Go
- Hosted on github
Simplified Cluster Management Stack

Cluster Scheduler

Scheduled Containers

Node Container Manager

Managed Base OS
Kubernetes Container Stack

- Kubernetes Master
- etcd
- Scheduled Containers
- kubelet + Docker
- Managed Base OS
Kubernetes

Container Agent
Machine Host

Container Agent
Machine Host

Container Agent
Machine Host

Container Agent
Machine Host

Container Agent
Machine Host

Container Agent
Machine Host

Container Agent
Machine Host

Container Agent
Machine Host

Kubernetes
Master/Scheduler
Pods

Web Server
Log Roller

Kubernetes Master/Scheduler

Container Agent
Container Agent
Container Agent
Container Agent
Container Agent
Container Agent
Container Agent
Container Agent

Machine Host
Machine Host
Machine Host
Machine Host
Machine Host
Machine Host
Machine Host
Machine Host

Google Cloud Platform
Too Many Pods

Kubernetes - Master/Scheduler

Container Agent
Machine Host

Container Agent
Machine Host

Container Agent
Machine Host

Container Agent
Machine Host

Container Agent
Machine Host

Container Agent
Machine Host

Container Agent
Machine Host

Container Agent
Machine Host

Container Agent
Machine Host

Container Agent
Machine Host
Labels

labels:
role: frontend
stage: production

Kubernetes - Master/Scheduler

<table>
<thead>
<tr>
<th>Container Agent</th>
<th>Container Agent</th>
<th>Container Agent</th>
<th>Container Agent</th>
<th>Container Agent</th>
<th>Container Agent</th>
<th>Container Agent</th>
<th>Container Agent</th>
</tr>
</thead>
</table>
Declarative Over Imperative

Imperative:
"for pod in pod{001..100} ; start $pod

Declarative:
"Run 100 copies of this pod with a target of <= 2 tasks down at any time"

Pros:
• Repeatable
• "Set it and forget it"
• Eventually consistent
• Easily updatable

Con:
• Tracing action/reaction can be difficult.
"I made a change, is it done?"
Replication Controller

- replicas: 4
- template:
  ...
- labels:
  role: frontend
  stage: production

Kubernetes - Master/Scheduler

- Container Agent
  - Machine Host
- Container Agent
  - Machine Host
- Container Agent
  - Machine Host
- Container Agent
  - Machine Host
- Container Agent
  - Machine Host
- Container Agent
  - Machine Host
- Container Agent
  - Machine Host
- Container Agent
  - Machine Host
Replication Controller

replicas: 1

template:

... 

labels:
role: frontend
stage: production
Replication Controller

replicas: 3

labels:
  role: frontend
  stage: production

Kubernetes - Master/Scheduler
Service

id: backend-service
port: 9000
labels:
  role: backend
  stage: production

Kubernetes - Master/Scheduler

Container Agent - Machine Host
Container Agent - Machine Host
Container Agent - Machine Host
Container Agent - Machine Host
Container Agent - Machine Host
Container Agent - Machine Host
Container Agent - Machine Host
Container Agent - Machine Host
Cat in a Container
Summing Up

1. We’re taking lessons we’ve learned and open sourcing them
2. Kubernetes is evolving
3. We’re eager to hear from you!
We are just getting started...

Clone Kubernetes at: 
[github.com/GoogleCloudPlatform/kubernetes](https://github.com/GoogleCloudPlatform/kubernetes)

IRC: #google-containers on Freenode

Google group: google-containers

Reach out:
  jbeda@google.com
  Twitter: @jbeda