Mesos vs Kubernetes
What We Learned Working With Both Customers
Sep 2017

Khalid Ahmed (khalida@ca.ibm.com)

Guang Ya Liu (liugya@cn.ibm.com)
Agenda

- Ecosystem of Container Cloud: Why Kubernetes and Mesos are Hot!
- Customer Journeys & Requirements
- Kubernetes & Mesos Comparisons
- Selecting Your Container Cloud
- IBM’s Choice: Kubernetes-based Container Cloud
Container Ecosystem

Layer 6
PaaS: Public, Private, Hybrid

Layer 5
CMP

Layer 4
Orchestration

Layer 3
Container Engine

Layer 2
Operating System

Layer 1
Infrastructure

Tool Chain
Why Container Cloud

HOW?

Applications

image1
image2
image3
...
imagen

service1, service2, service3 ... servicen
Basic Requirement

**Container Cloud**

- SaaS
- PaaS
- IaaS

**Container Orchestration**

- Resource Management

**External Service Integration**

- Heterogeneous Resource
- App Center
- Auto Scaling
- Load Balance
- Multi Tenant
- Image Management
- Alarm/Monitor/Log
- CI/CD
- Unified UI
Other Requirement

- **Self Control**
  - Based on Open Source
  - No vendor lock-in

- **Ecosystem**
  - Services
  - Developers
  - Users

- **HA**
  - Platform
  - Application

- **Scalability**
Main Solutions for Container Cloud

- **Open Source Projects**
  - ✔ Mesos + Marathon
  - ✔ Kubernetes

- **Target**
  - ✔ Distributed System Management
  - ✔ Scalability and Customization
  - ✔ Resource Management
  - ✔ Application Life Cycle Management
Kubernetes Architecture

Many components are pluggable:
- schedulers
- container runtime
- persistent data store
- cloud providers
- ...
Mesos Architecture
Kubernetes + Mesos Solution

- Long Running Services
- Short Tasks
- Big Data
- Streaming

Unified UI
AuthN/AuthZ
App Center

Resource Scheduler
Heterogeneous Resource
HA

SDN (DNS, Overlay)
SDS (Ceph, Spectrum Scale)

IaaS (OpenStack/VMWare)
## Main Solutions for Container Cloud

<table>
<thead>
<tr>
<th></th>
<th><strong>Mesos (DC/OS)</strong></th>
<th><strong>Kubernetes</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Area</strong></td>
<td>Internet, Finance,</td>
<td>Telecom, Bank, Insurance</td>
</tr>
<tr>
<td><strong>Scalability</strong></td>
<td>Big (&gt;5000 nodes in a single cluster)</td>
<td>Not that big (&lt;=5000 nodes in a single cluster, but have federation cluster to support large scale cluster)</td>
</tr>
<tr>
<td><strong>DevOps</strong></td>
<td>Need a specific team to maintain as sometimes need to customize the mesos framework for applications.</td>
<td>Need less people to maintain, have a large and open ecosystem</td>
</tr>
<tr>
<td><strong>Example Customer Cases</strong></td>
<td>Twitter, Apple, Airbnb, Uber etc</td>
<td>Github, JingDong, Qunar etc</td>
</tr>
<tr>
<td><strong>Focus</strong></td>
<td>Resource Management, but using Marathon for Application Orchestration</td>
<td>Application Orchestration, but also enhancing Kubernetes for resource management such as preemption etc</td>
</tr>
<tr>
<td><strong>Infrastructure</strong></td>
<td>Supports both container and OS process level</td>
<td>Containers</td>
</tr>
</tbody>
</table>
## Open Source Community (Till 2017.7)

<table>
<thead>
<tr>
<th></th>
<th>Mesos (DC/OS)</th>
<th>Kubernetes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contributors</td>
<td>346</td>
<td>2505</td>
</tr>
<tr>
<td>Companies</td>
<td>113</td>
<td>789</td>
</tr>
<tr>
<td>Time Zone</td>
<td>9</td>
<td>15</td>
</tr>
<tr>
<td>Total Commits</td>
<td>12400+</td>
<td>49000+</td>
</tr>
<tr>
<td>Commits per day</td>
<td>3-10</td>
<td>41+</td>
</tr>
<tr>
<td>Release Cycle</td>
<td>60 days</td>
<td>100 days</td>
</tr>
<tr>
<td>Github fork</td>
<td>1270+</td>
<td>8300+</td>
</tr>
<tr>
<td>Github Star</td>
<td>3150+</td>
<td>23600+</td>
</tr>
<tr>
<td>Meetups</td>
<td>200+</td>
<td>200+</td>
</tr>
</tbody>
</table>

Source: [Link to presentation](https://speakerd.s3.amazonaws.com/presentations/673edc72242c413485ec0c8c44d892d5/Kube_Leadership_Summit_2017_-_State_of_the_Project.pdf)
How are container and PaaS tools used in the past 12 months?

Since there are relatively few answers to the previous question, “Which container and PaaS tools are used to manage applications?” (192 in April 2017 and 118 in April 2016), and the amount of crossover among those answering both surveys was relatively low (about a third), we also looked at the combined answers for both April 2016 and April 2017, while de-duplicating those who answered in both surveys (keeping their most recent answer). Kubernetes leads, followed by OpenShift and CloudFoundry.


Figure 5.5 n=282

- Production
- Dev/ QA
- Proof of Concept
Solution Selection

- **Mesos**
  - ✓ Already have some workload running, such as Hadoop, Spark, Kafka etc and want to share resources
  - ✓ Large Scale
  - ✓ Both container and non-container environment

- **Kubernetes**
  - ✓ New applications
  - ✓ Medium scale single cluster/ Federated cluster
  - ✓ Only containers

- **Kubernetes + Mesos**
  - ✓ Innovation and research projects
  - ✓ Investigate how to leverage advantages from both Kubernetes and Mesos
Example Use Case: Financial Services Company

System of Engagement (Web)
- Red Packet
- Customer Service
- Flow Control
- Mobile Banking
- Online Banking
- WeChat Bank
- Marketing Content Management
- Business Operations Management

System of Insight (BigData)
- Analysis of Marketing
- Business Management
- Business Service

Containerized Apps
- Online Services
- System Services

Non-Containerized Apps
- Oracle

Infrastructure
- X86
- POWER

Scalability
- 1000 nodes

Business Operations Monitoring
- Audit
- Log Analysis
- Middleware Monitor

Kubernetes

SDN: Host/Bridge/Overlay
Power

SDS: Ceph, Spectrum Scale
X86

WebSphere
ELK
Prometheus
Alert Manager
Nginx
Redis
Oracle
Jenkins
App Store
Why Kubernetes in IBM?

- Kubernetes governance model – part of Linux Foundation
- A growing and vibrant ecosystem – multiple cloud vendors (Amazon, Google, Microsoft, etc) participate avoiding vendor lock-in
- IBM has proven experience on operating Kubernetes clusters at scale (Bluemix Container Service, IBM GTS)
- IBM is Kubernetes top contributor (No. 9) / Platinum member of the CNCF initiative
- IBM is one of the leaders for Istio which is an important add-on service mesh manager for Kubernetes
- IBM is leading two incubator project in Kubernetes
  - Service catalog [https://github.com/kubernetes-incubator/service-catalog](https://github.com/kubernetes-incubator/service-catalog)
  - kube arbitrator [https://github.com/kubernetes-incubator/kube-arbitrator](https://github.com/kubernetes-incubator/kube-arbitrator)
IBM Bluemix Container Service

Carrier Master
- kubelet
- etcd
- kube-scheduler
- kube-apiserver
- kube-controller-manager
- Docker

IBM Bluemix Container Service Control Plane

Cruiser Master
- etcd
- kube-scheduler
- kube-apiserver
- kube-controller-manager
- Docker

Customer’s Infrastructure Account
- kubelet
- Customer pods
- kube-dashboard
- kube-dns
- kube-proxy
- Docker

K8s master node

K8s worker nodes
IBM Cloud private

Free to Download and Use
https://hub.docker.com/r/ibmcom/cfc-installer/

Kubernetes based container platform
Industry leading container orchestration platform across private, dedicated & public clouds

Common Services
To simplify operations management, DevOps, and hybrid integration

IBM Middleware, Data & Analytics Services
Cloud enabled middleware, application runtimes, messaging, databases & analytics to optimize current investments and rapidly innovate

Cloud Foundry
For prescribed application development & deployment

With flexible infrastructure support: install on Baremetal or OpenStack or VMware

Managed by customer or by IBM
Thank You