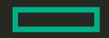
Container Orchestration: Which Conductor?

ContainerCon Europe, Berlin, Oct 2016



Mike Bright, @mjbright

Haikel Guemar, @hguemar

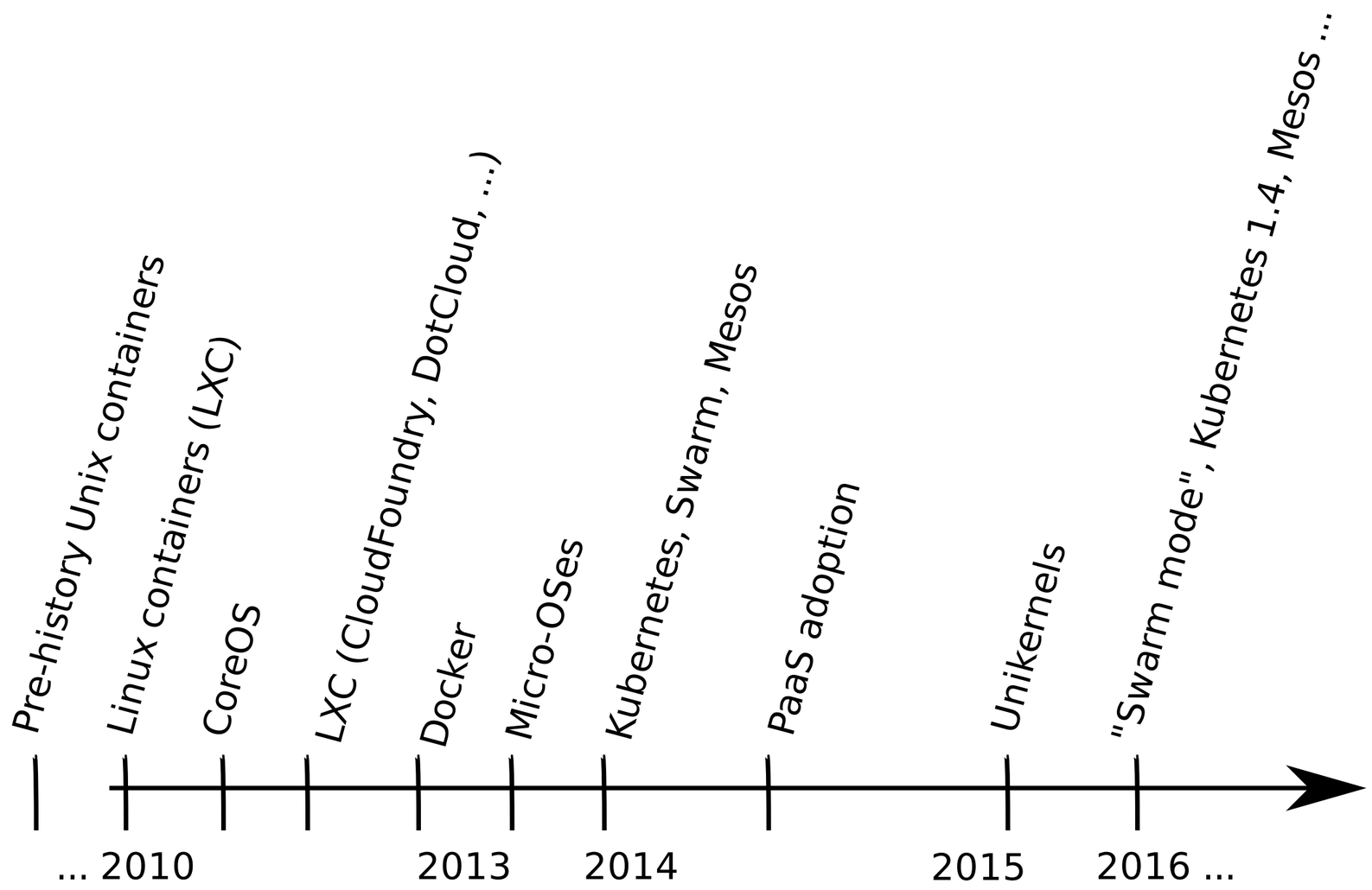
Mario Loriedo, @mariolet

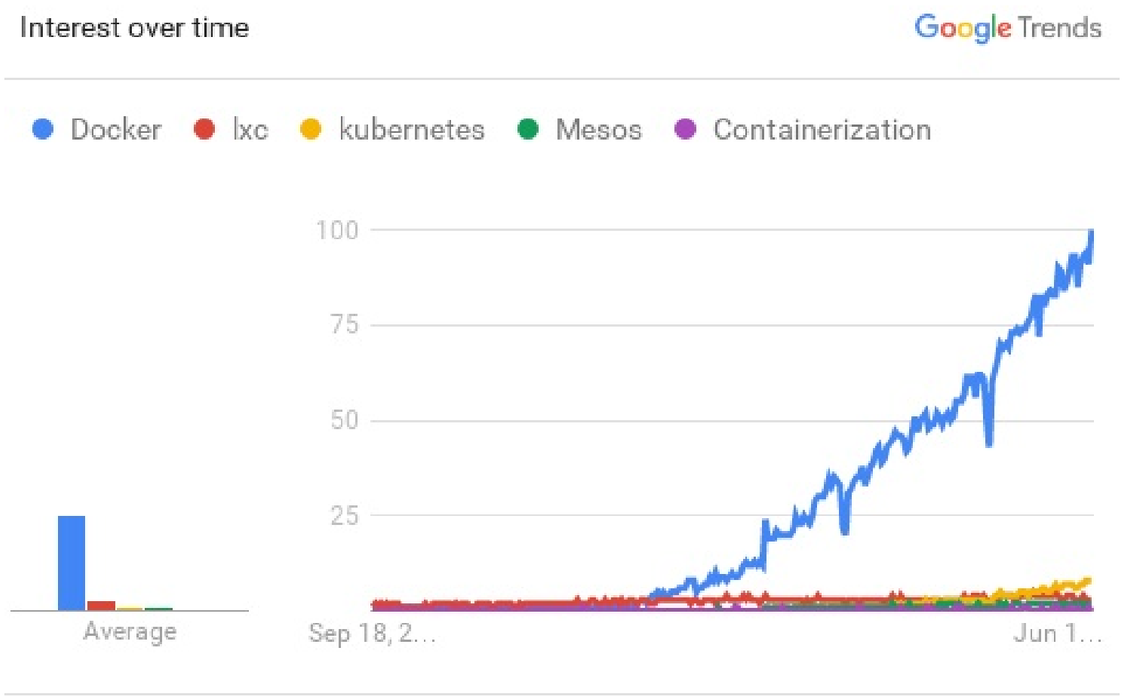
First ...

First ...

A little bit of history

So let's first look at recent container history …




Interest over time
Google Trends
Docker
lxc
kubernetes
Mesos
Containerization
100
75
50
25
Average
Sep 18, 2...
Jun 1...
Worldwide. Past 5 years.

µ-OSes

Many vendors are developing µ-OSes, small OS (mainly Linux-based) to be the basis for container engine hosts whether they be bare-metal or virtual **host machines**.

They're small, with fast startup, use few resources and have a small attack surface and often *"atomic"* software updates.

OS	**Vendor**
CoreOS	- (CoreOS)
Project Atomic	- (RedHat)
Photon	- (VMWare)
RancherOS	- (Rancher Labs)
Nano Server OS	- (Microsoft)
Ubuntu Snappy Core	- (Canonical)

µ-OSes

Many vendors are developing µ-OSes, small OS (mainly Linux-based) to be the basis for container engine hosts whether they be bare-metal or virtual **host machines**.

They're small, with fast startup, use few resources and have a small attack surface and often *"atomic"* software updates.

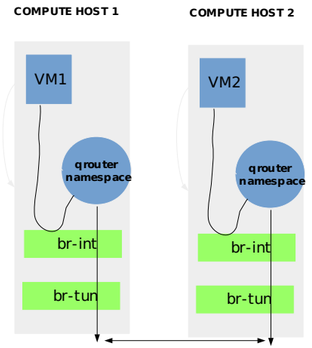
OS	**Vendor**
CoreOS	- (CoreOS)
Project Atomic	- (RedHat)
Photon	- (VMWare)
RancherOS	- (Rancher Labs)
Nano Server OS	- (Microsoft)
Ubuntu Snappy Core	- (Canonical)

- …Unikernels (…)

µ-Services

From monoliths to μ-services

Remember when **high availability** meant this …?



Servers running **monolithic applications** in **Active-Standby** modes, as 1+1, N+1, or N+M or split across 3 tiers.

Scaling meant to **"scale up"** by adding CPU, RAM, disk. But there's a limit to this … then you have to **"scale out"**

µ-services

From monoliths to µ-services

Then came µ-services ..

As the industry moved to virtualized micro-services this allowed to obtain greater efficiencies (higher utilisation of resources) and the redesign of applications allows to scale out and achieve high availability.

Containers facilitate this move, allowing faster scaling and even greater efficiencies with less redundancy (no OS to reproduce).

µ-services

How containers help?

Container solutions such as Docker go beyond the isolation capabilities of LXC by providing simple to use tools to enable packaging of apps with their dependencies allowing portable applications between systems.

Containers are lightweight

Containers can be shared

Containers allow to use the same application binaries on development, test and production systems whether that be on a laptop, server or in the cloud.

From monoliths to µ-services

But 1000's of nodes are unmanageable … aren't they?

We can't take care of our  ,

so we have to treat them like 

that's cloud native !

So we need container orchestration

Orchestration

What was Container Orchestration again?

… and how does it differ from automation?

Orchestration is

- Architecture
  - Composition
  - Stitching
- Workflows
- Policies

Orchestration

- We already have many choices for

Container Orchestration Engines

Docker Swarm	: Docker Inc.
Kubernetes	: Cloud Native Computng Foundation
Apache Mesos	: Apache Software Foundation
Fleet	: CoreOS
Rancher	: Rancher Labs
Nomad	: HashiCorp

Orchestration

Imperative or Declarative

To manage 1000's of nodes it becomes necessary to be able to make declarative requests to the system.

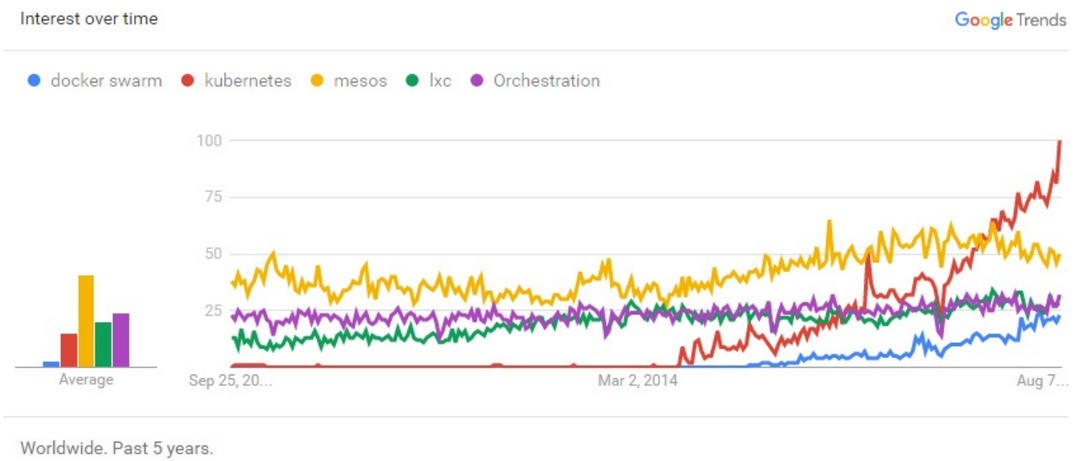
	Imperative	**Declarative**
Tell the system	what to do	desired state
	"start a new node"	*"3 mysql nodes"*
Intelligence	Operator	Orchestration Engine
	…	
Flexibility	Best	Least

Choice is great - when you know what you want ...

The Big 3 - Main Orchestration Choices

- Docker Swarm
  - Docker swarm
  - The swarm toolkit
  - Docker **"swarm mode"**
- Apache Mesos
  - Frameworks
    - Marathon (Mesosphere), Chronos, Aurora
  - Plugins
    - Jenkins
  - Mesosphere, DC/OS
- Kubernetes

The Big 3 - What does Google Trends say?



Clearly Kubernetes has a lead, but we can expect *"Docker Swarm"* to quickly make progress thanks to the new *"swarm mode"*

But lets not forget the alternatives …

More Choice ...

- Rancher (Rancher Labs)
- Fleet (CoreOS) A distributed init system (between systemd and etcd)
- Nomad (HashiCorp)

Docker Swarm

Docker Swarm

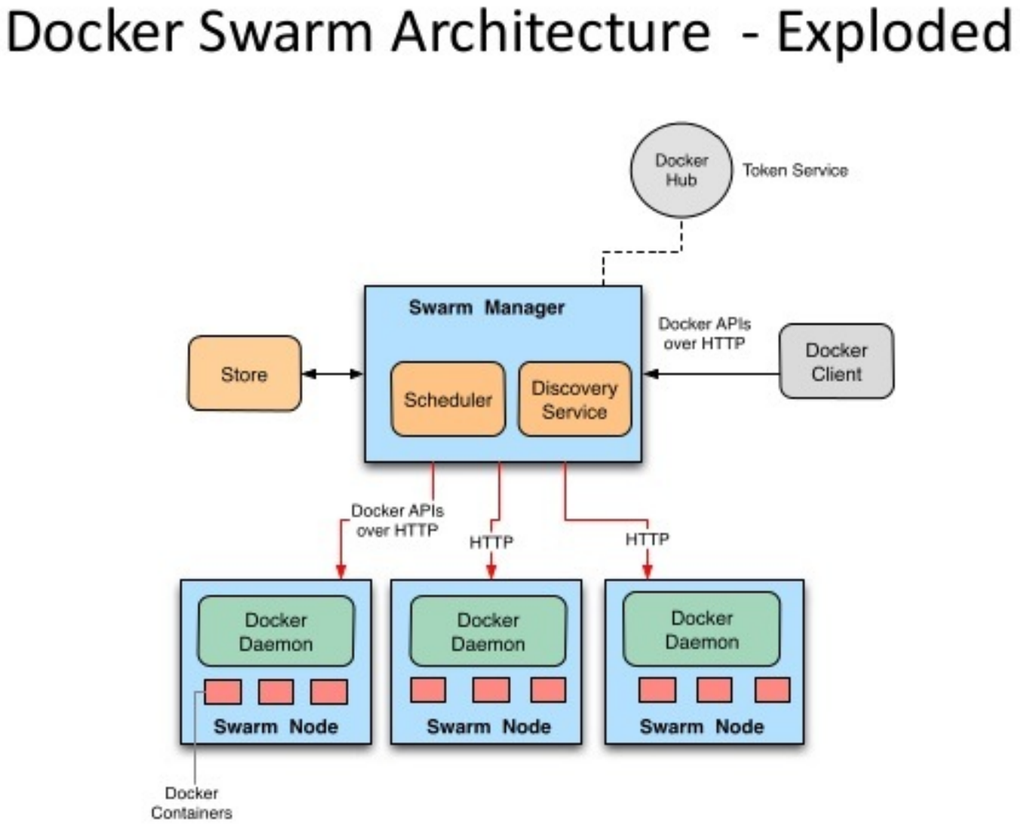
- 2014 Dec: Docker Swarm was announced
  - Capable of scaling up an architecture defined by Docker Compose
- 2015: Swarm Toolkit released
- 2015 June: Docker engine with Swarm Mode released with Docker 1.12

Docker "Swarm Mode" is quite a revolution in Docker Engine capabilities as it integrates

- Integration of Orchestration Capabilities directly within the Docker Engine
  - Load balancing across a mesh network
  - Service Discovery
- The same ease of use we're accustomed to from Docker !

Architecture

Docker Swarm



Using Docker "Swarm Mode"

Docker Swarm

Create a new swarm by creating the master node:

```
$ docker swarm init
xxxx
```

Join a new node to the swarm swarm:

```
$ docker swarm join ....
```

Getting started

Docker Swarm

An excellent place to start is with Jerome Pettazoni's *"Orchestration Workshop"*

- being run at this conference
- Available on github, https://github.com/jpetazzo/orchestration-workshop

Apache Mesos

Apache Mesos

Arguably the most production ready orchestration today, exists since 2009.

Can scale to ~ 10,000 nodes.

Used in production by:

- Twitter
- Groupon
- Netflix

Mesos is used in conjunction with Frameworks such as

- **Marathon** or Aurora: manages long running tasks
- Chronos: designed for job orchestration
- Hadoop: for big data processing
- Kubernetes: allowing delcarative use

mesos.apache.org

MESOS

Apache Mesos

Architecture

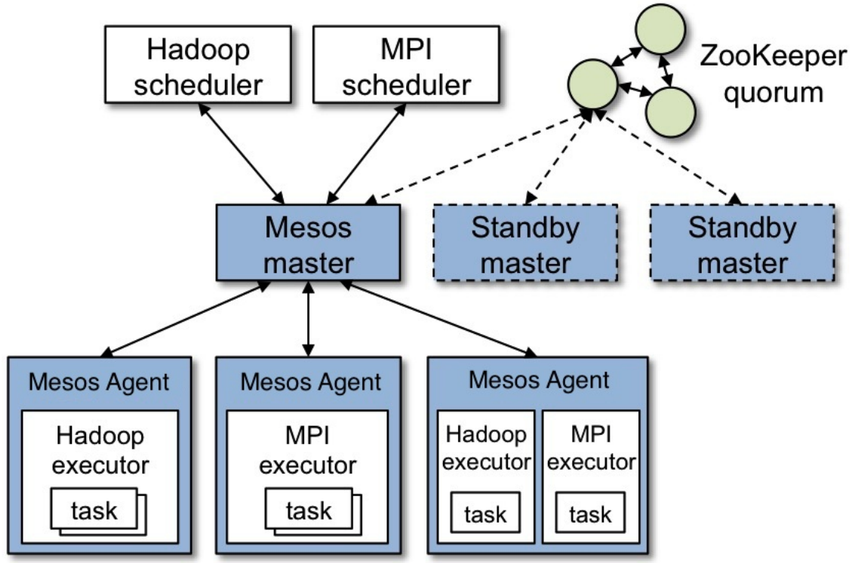


Image courtesy of
http://mesos.apache.org/documentation/latest/architecture/

mesos.apache.org

Using Mesos

Apache Mesos

Step1

xxxx

Step2

Apache Mesos

Getting started

An excellent place to start is with the following tutorials

- Mesos
  - minimesos
- Mesosphere

Kubernetes

From the Greek: "Steersman, helmsman, sailing master"

Kubernetes

Kubernetes is an open source project created by Google based on it's extensive experience running containers (millions of containers over a decade or so) from it's Borg and Omega projects.

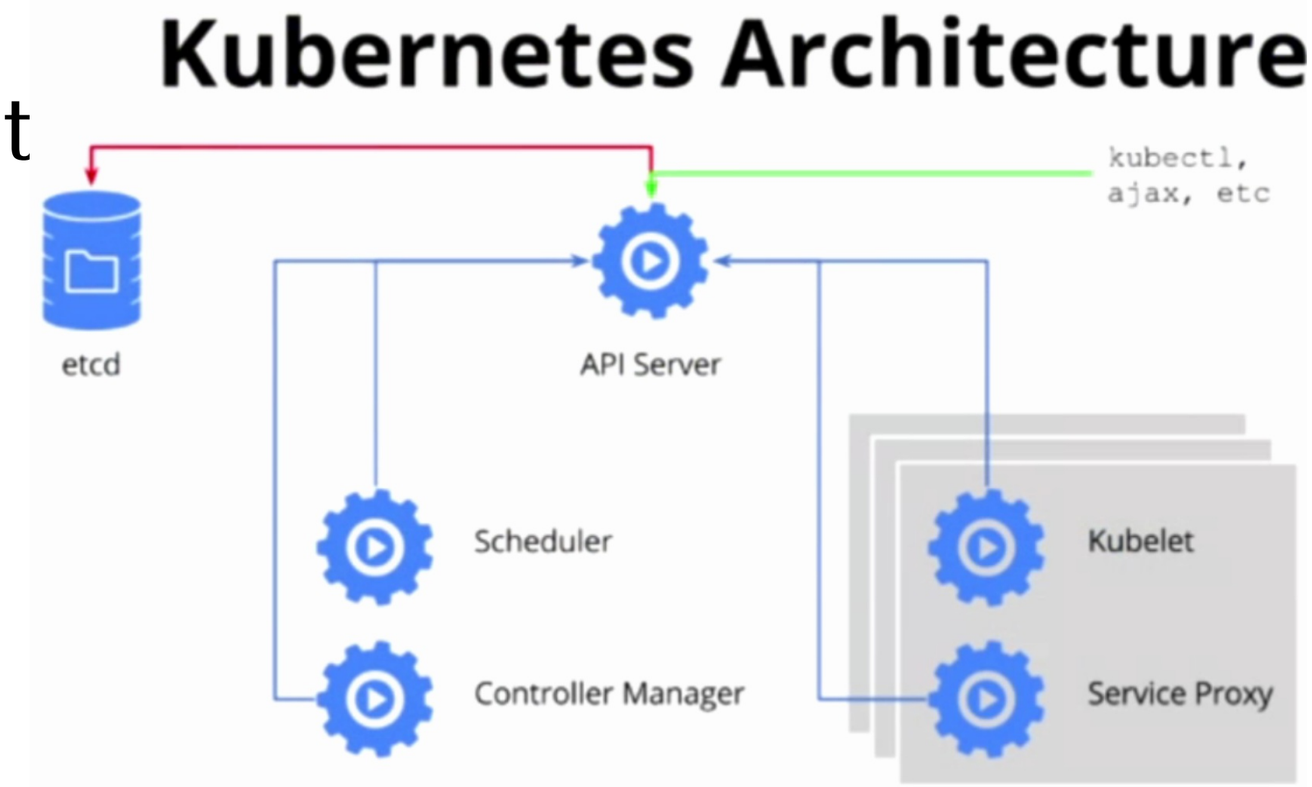
Started ~ Oct 2014, reach v1.0 in in July 2015 and currently at v1.4 It is managed by the Cloud Native Computing Foundation https://cncf.io/

Integrated in:

- Stackanetes, Mirantis
- RedHat OpenShift PaaS
- Deis Paas
- EBay : Kubernetes + OVS
- CoreOS: Tectonic (commercial Kubernetes offering)

Architecture

Kubernet



Concepts

- Cluster
- Node
- Pod
- Replication controller
- Service
- Label

Using Kubernetes

Step1

Kubernetes xxxx

Step2

Getting started

An excellent place to start is with the following tutorials

Kubernetes:

- minikube?
- ...

Industry Players

Choices made by Industry Players

- RedHat: Completely redesigned their OpenShift PaaS to use Docker Containers and Kubernetes, and created Project Atomic
- CoreOS: CoreOS, created the company 6 months after Docker was announced with a goal of providing **GIFFE**
- Google: Kubernetes used for GCP
- MicroSoft: Commited to port Docker to Windows (Windows Server 2016, Azure)
- VMWare

So isn't it time we told you what to choose?

So isn't it time we told you what to choose?

... well we'll provide some guidelines at least ...

What's common

They are converging on many points

They are tending to add 'declarative specification' capabilities.

It's no longer feasible for an operator to decide on which node to deploy especially when complex constraints exist

- making use of specialized hardware, e.g. SSD best for some operations
- adapting to hardware failures

An operator specifies the "desired state" and the orchestrator does the rest.

What's different

Rancher lightweight

Comparison

Feature	Swarm	Kubernetes	Mesos
Declarative		Yes	

Rancher lightweight

Hands on ...

@hguemar @mjbright @mariolet

Hands-on

Come along

This afternoon's tutorial session led by Mario:
Tuesday, October 4 - 15:30 - 16:20

5 Containers for 5 Languages: Patterns for Software Development Using Containers - Mario Loriedo, Red Hat

Tomorrow's lab session led by Haikel:
Wednesday, October 5 - 11:00 - 12:50

Container Orchestration Lab: Swarm, Mesos, Kubernetes - Haïkel Guémar, Fedora Project

Lab setup instructions here

- Docker Swarm
- Kubernetes
- Apache Mesos

Docker Swarm Demo

Questions?

Thank you

Resources

@hguemar @mjbright @mariolet

Resources

Books

Publisher	Title	Author
OReilly	Docker Cookbook	Sébastien Goasguen
OReilly	Docker Up & Running	Karl Matthias, Sean P. Kane
OReilly	Using Docker	Adrian Mouat
OReilly	[Early Access] Kubernetes Up & Running	Kelsey Hightower
Manning	[MEAP] CoreOS in Action	Matt Bailey
Manning	[MEAP] Kubernetes in Action	Marko Lukša

Resources

Articles/Organisms

Cloud Native Computing Foundation - Kubernetes, Prometheus https://cncf.io/

"Kubernetes the Hard Way, Kelsey Hightower" - https://github.com/kelseyhightower/kubernetes-the-hard-way *"Kubernetes User Guide, Walkthrought"* - http://kubernetes.io/docs/user-guide/walkthrough/

Resources

Videos

- June 2016 - Container Orchestration Wars, Karl Isenberg, Mesosphere
- Mar 2016 - Container Orchestration with Kubernetes, Docker Swarm & Mesos-Marathon - Adrian Mouat, Container Solutions
- Jan 2016 - Docker, Kubernetes, and Mesos: Compared.,,Adrian Otto, Southern California Linux Expo

Repos

Documentation

Kubernetes

- **Getting started guides**
  - Creating a Kubernetes Cluster
  - port Kubernetes to a new environment
    - in Getting Started from Scratch
- **User documentation**
  - to run programs on an existing Kubernetes cluster
    - Kubernetes User Guide: Managing Applications
  - the Kubectl Command Line Interface is a detailed reference on the `kubectl` CLI
  - User FAQ

Kubernetes

Documentation - 2

- **Cluster administrator documentation**
  - for people who want to create a Kubernetes cluster and administer it
  - in the Kubernetes Cluster Admin Guide
- **Developer and API documentation**
  - to write programs using the Kubernetes API, write plugins or extensions, or modify core code
  - Kubernetes Developer Guide
  - notes on the API
  - API object documentation, a detailed description of all fields found in the core API objects
- **Walkthroughs and examples**
  - hands-on introduction and example config files
  - in the user guide
  - in the docs/examples directory
- **Contributions from the Kubernetes community**
  - in the docs/contrib directory

Documentation 3

Kubernetes

- **Design documentation and design proposals**
  - to understand the design of Kubernetes, and feature proposals
  - Kubernetes Design Overview and the docs/design director y
  - docs/proposals directory
- **Wiki/FAQ**
  - the wiki
  - troubleshooting guide

Community, discussion, contribution, and support

Consider joining the Cloud Native Computing Foundation. For details ab out who's involved and how Kubernetes plays a role, read their announcement.