

# Container Orchestration: Which Conductor?

ContainerCon Europe, Berlin, Oct 2016



**Hewlett Packard**  
Enterprise

Mike Bright,  @mjbright



**redhat.**

Haikel Guemar,  @hguemar



**redhat.**

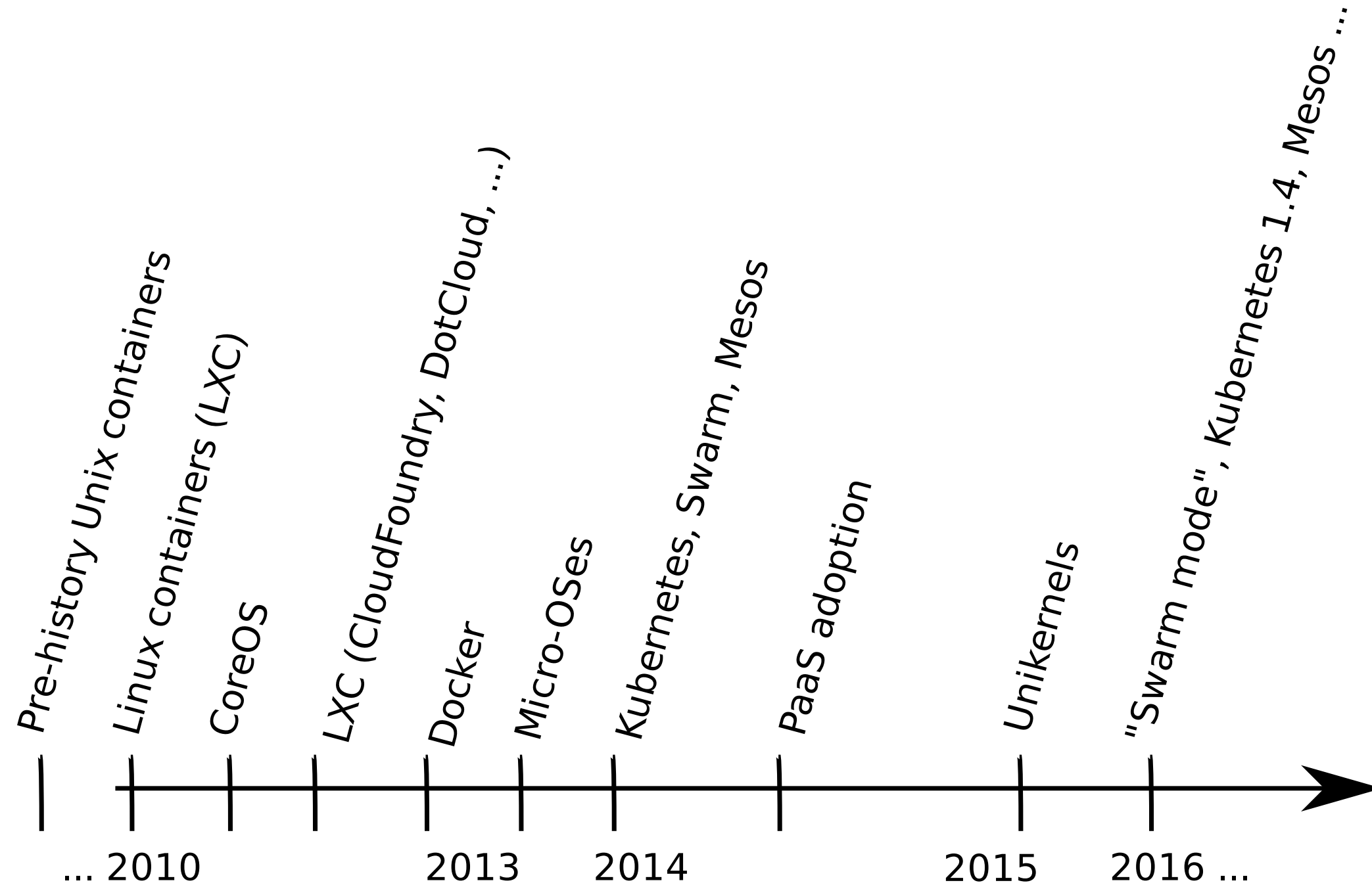
Mario Lorio,  @mariolet

First ...

First ...

A little bit of history

So let's first look at recent container history ...

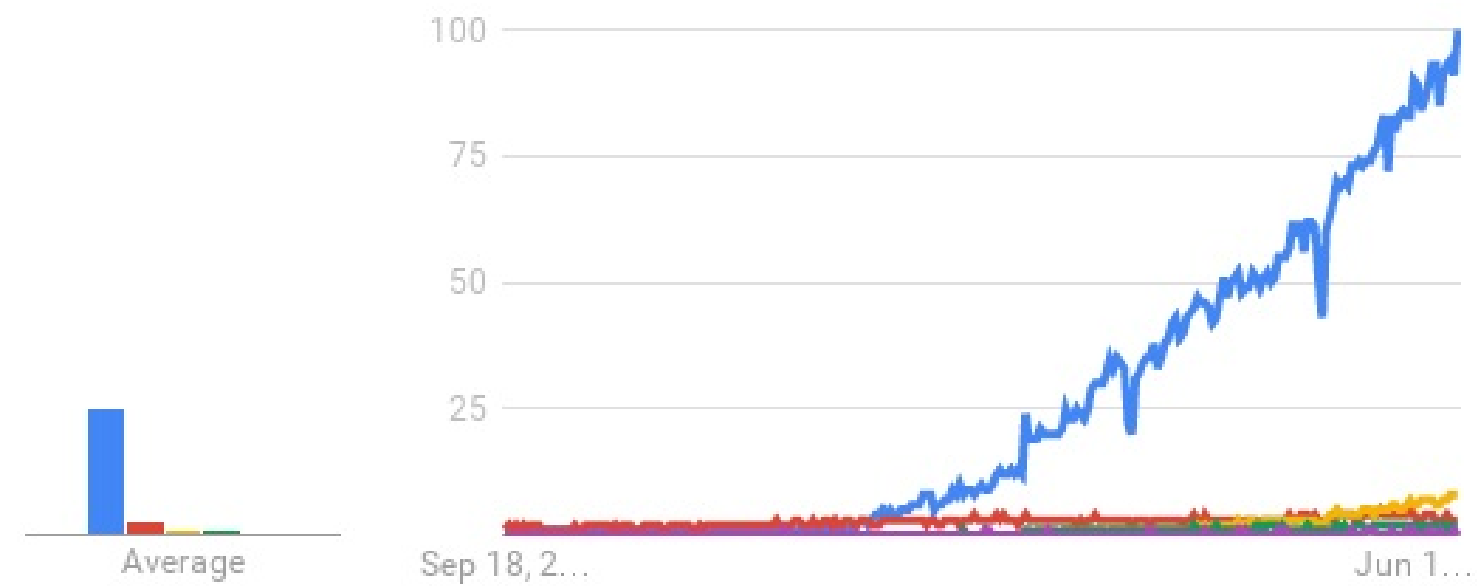


@hguemar @mjbright @mariolet

Interest over time

Google Trends

● Docker ● lxc ● kubernetes ● Mesos ● Containerization



Worldwide. Past 5 years.

@hguemar @mjbright @mariolet

# History

## 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**.

These OSes are small, with fast startup, deployment, 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)

# History $\mu$ -OSes

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

These OSes are small, with fast startup, deployment, 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 (...)@hguemar @mjbright @mariolet

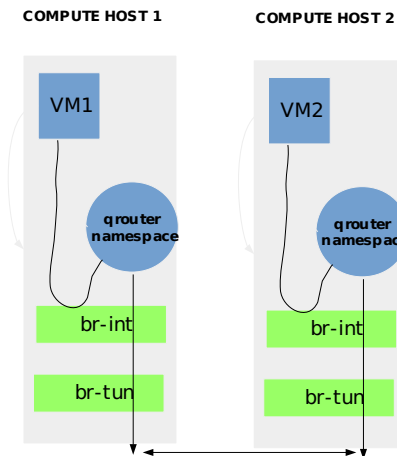
# $\mu$ -Services



μ-  
services

# From monoliths to μ-services

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



Server 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**

# $\mu$ -services      From monoliths to $\mu$ -services

**Then came  $\mu$ -services ..**

Now we can achieve much better hardware utilisation because of the smaller size of components.

@hguemar @mjbright @mariolet

μ-  
services

# 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 !

@hguemar @mjbright @mariolet

# 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.

@hguemar @mjbright @mariolet

So we need container orchestration

# Orchestration

.we already have many choices for  
**Container Orchestration Engines**

Docker Swarm : Docker Inc.

Kubernetes : Cloud Native Computing Foundation

Apache Mesos : Apache Software Foundation

Fleet : CoreOS

Rancher : Rancher Labs

Nomad : HashiCorp

These COEs are to varying degrees  
**Imperative** or **Declarative**

@hguemar @mjbright @mariolet

# Orchestration Imperative or Declarative

	<b>Imperative</b>	<b>Declarative</b>
<b>Tell the system</b>	what to do  <i>"start a new node"</i>	desired state  <i>"3 mysql nodes"</i>
<b>Intelligence</b>	Operator  ...	Orchestration Engine
<b>Flexibility</b>	Best	Least

@hguemar @mjbright @mariolet

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



# Orchestration

## The Big 3 - Main Orchestration Choices



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

But lets not forget the alternatives ...

@hguemar @mjbright @mariolet

# Orchestration More Choice ...



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

The following are static configuration engines which can be used to automate tasks but they are not orchestration engines as such:

- Ansible
- CloudSlang
- Vagrant
- Juju

# Docker Swarm





# Docker Swarm

- Docker engine with Swarm Mode
- Swarm Toolkit
- Swarm

Used in production by:

- ???
- .... ????

Integrated in:

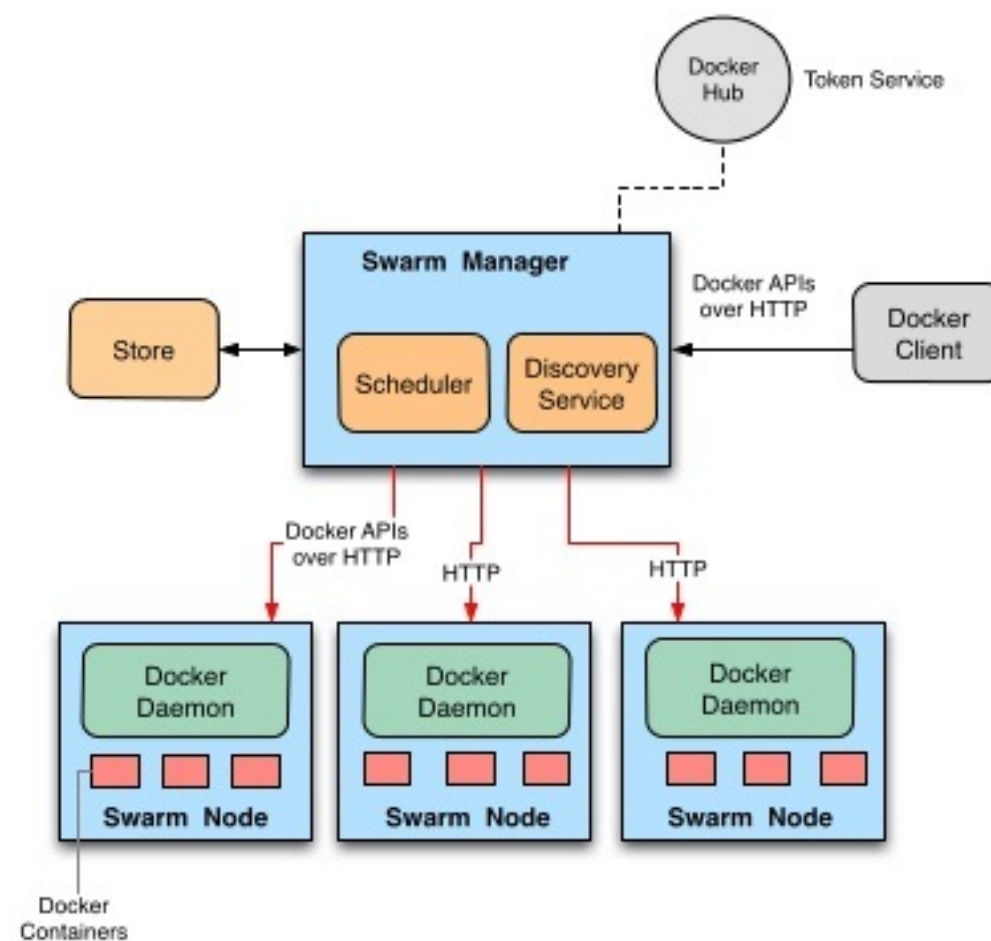
- OpenStack Magnum Project; one of the supported COEs



# Architecture

## Docker Swarm

### Docker Swarm Architecture - Exploded





# Docker Swarm

## Getting started

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
- .... ????

Integrated in:

- OpenStack Magnum Project; one of the supported COEs

Mesos is used in conjunction with Frameworks such as

- Marathon: manages long running tasks
- Chronos: designed for job orchestration
- Hadoop: (YARN?)
- Kubernetes: allowing declarative use

[mesos.apache.org](http://mesos.apache.org)





Apache  
Mesos

# Architecture

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

[mesos.apache.org](http://mesos.apache.org)



# 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 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.

# Kubernetes

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/>

Used in production by:

- ???
- .... ????

Integrated in:

- Stackanetes, Mirantis
- OpenShift
- Deis <http://deis.io>
- EBay : Kubernetes + OVS
- CoreOS: Tectonic (commercial Kubernetes offering)
- OpenStack Magnum Project; one of the supported COEs

[kubernetes.io](https://kubernetes.io)

@hguenard @mjbright @mariolet



# Architecture

## Kubernetes

[kubernetes.io](https://kubernetes.io) @hguemar @mjbright @mariolet



# Concepts

- Cluster

## Kubernetes

- Node
- Pod
- Replication controller
- Service
- Label



# Getting started

An excellent place to start is with the following tutorials

Kubernetes <sup>minikube?</sup>  
• ...

# Industry Players



# 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: Committed to port Docker to Windows (Windows Server 2016, Azure)
- VMWare

@hguemar @mjbright @mariolet

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 Rancher lightweight  
different

@hguemar @mjbright @mariolet

# Comparison

Feature	Swarm	Kubernetes	Mesos
Declarative		Yes	

Rancher lightweight

@hguemar @mjbright @mariolet

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

@hguemar @mjbright @mariolet



## Demo - Apache Mesos

@hguemar @mjbright @mariolet

## Demo - Kubernetes

@hguemar @mjbright @mariolet

## Demo - Docker Swarm

@hguemar @mjbright @mariolet

# Questions?

# Thank you

@hguemar @mjbright @mariolet

# 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

@hguemar @mjbright @mariolet

# 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, Walkthrough"* - <http://kubernetes.io/docs/user-guide/walkthrough/>

@hguemar @mjbright @mariolet

# 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

@hguemar @mjbright @mariolet





# 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](#)



## Documentation - 2

# Kubernetes

- **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 directory](#)
  - [docs/proposals directory](#)
- **Wiki/FAQ**
  - the [wiki](#)
  - [troubleshooting guide](#)

### **Community, discussion, contribution, and support**

Consider joining the [Cloud Native Computing Foundation](#). For details about who's involved and how Kubernetes plays a role, read [their announcement](#).