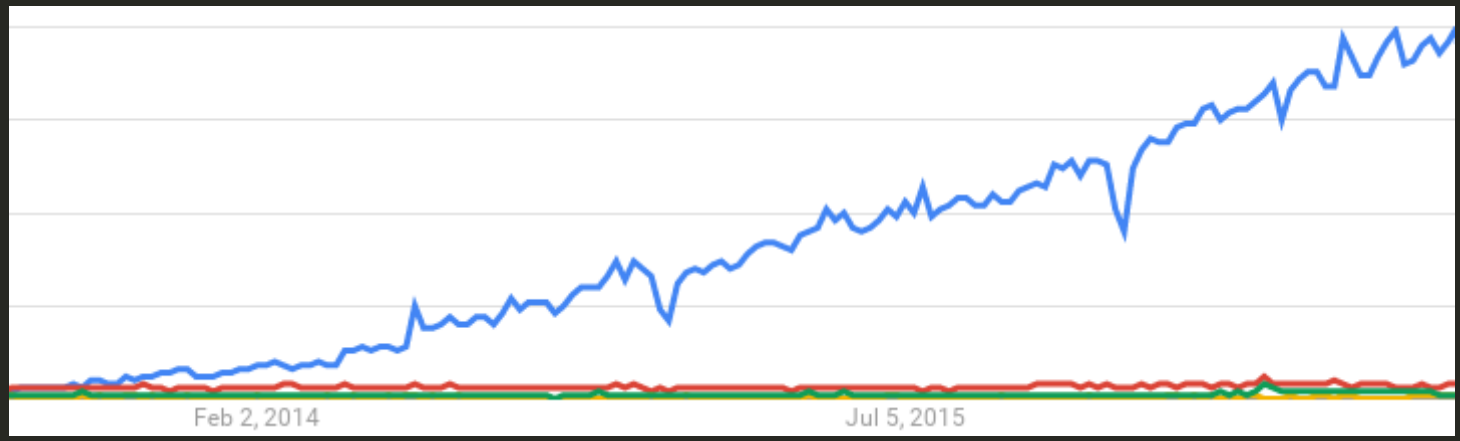
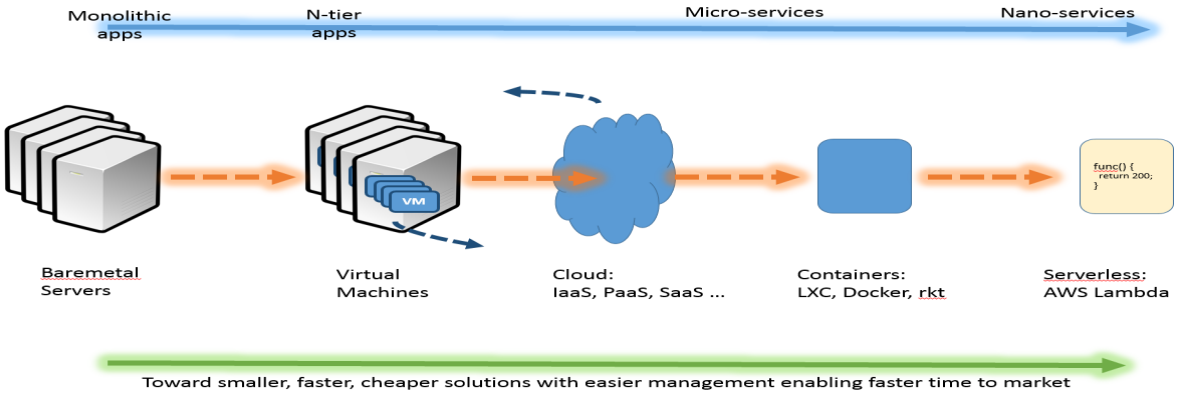
Hewlett Packard
Enterprise


Feb 2, 2014
Jul 5, 2015

The evolution from Monoliths to μ-Services

From monoliths to μ-services



Container-OSes

Vendors are developing Container-OSes, small OS (mainly Linux-based) for container engine hosts (bare-metal or virtual **host machines**).

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

OS	Vendor
Alpine Linux	- ()
Container Linux	- (CoreOS)
Project Atomic	- (RedHat)
RancherOS	- (Rancher Labs)
Photon	- (VMWare)
Nano Server OS	- (Microsoft)
Ubuntu Snappy Core	- (Canonical)
MicroOS	- (Suse)

...Unikernels

From monoliths to µ-services

But 1000's of instances are **unmanageable** ...

We can't take care of our pets  ,

so we have to treat them like cattle 

that's cloud native !

So we need Container Orchestration

What was Container Orchestration again?

Cluster management: Distribution of tasks across a cluster of nodes such as a DataCenter.
Placement & Routing: Task placement, interconnection and scaling to achieve fault tolerance, optimisation and load balancing
Monitoring: Detecting faults through health/liveness checks, monitoring resource utilisation
Resources: Manage compute, network and storage resources
Other: Logging, secrets management, rolling updates ...

Orchestration: Getting to *"Desired State"*

To manage 1,000's of instances we need to express *"desired state"* rather than *"do this"*.

	Imperative	**Declarative**
Tell system	Do this *"start a new instance"*	desired state *"3 x mysql"*
Intelligence	Operator	Orchestration Engine
Flexibility	Best	Least

It is no longer feasible for an operator to

- know the resources available on each node (e.g. SSD/HDD, GPU, ...)
- react to failure, know when to scale ...

@mjbright

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

The Orchestration Choices

- Docker Swarm ("Swarm Mode")
- Apache Mesos
- Kubernetes

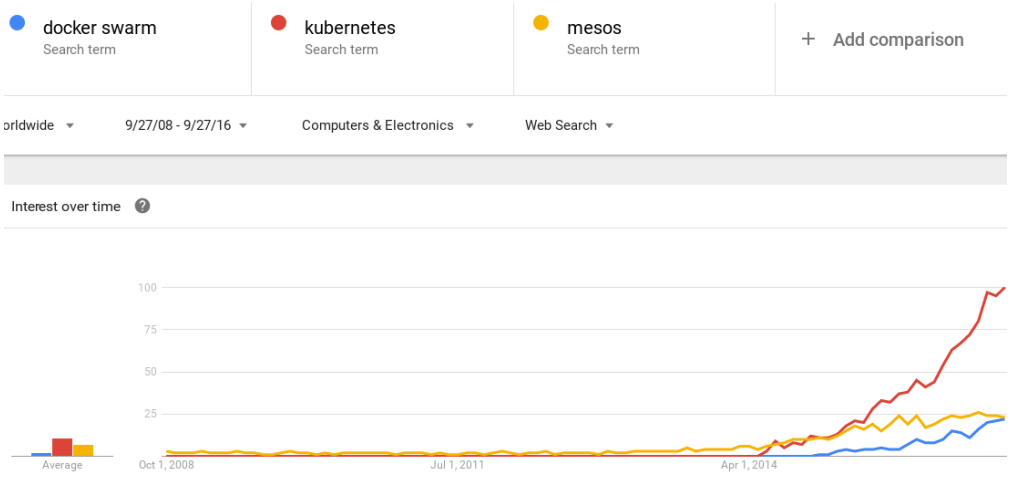
The Orchestration Choices

- Docker Swarm ("Swarm Mode")
- Apache Mesos
- Kubernetes

... more Choices ...

- Nomad (HashiCorp)
- Kontena (Kontena)
- Cattle (Rancher Labs)
- Fleet (CoreOS)
- Cloud Providers (AWS ECS, Azure ACS, Google GKE)

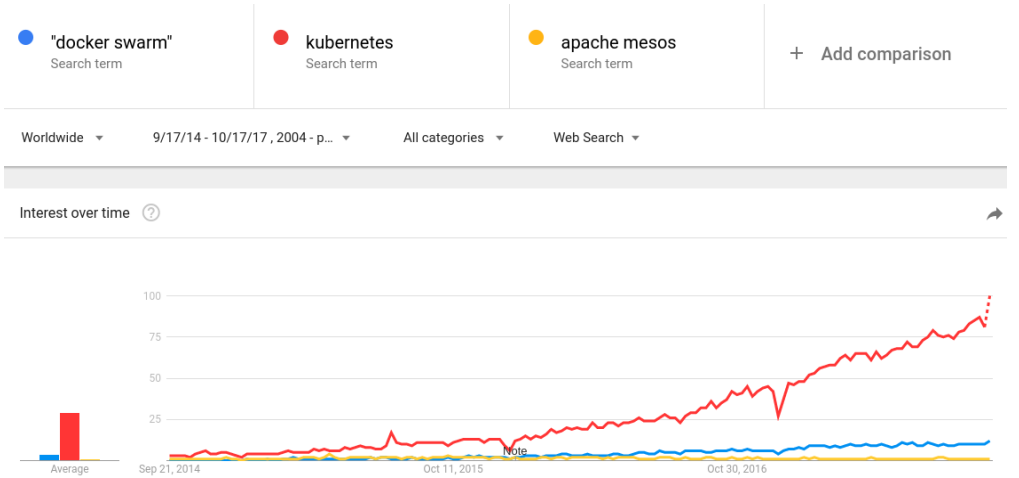
The Big 3 - Google Trends (2016)



Clearly Kubernetes has a lead in *Google "search trends"*

But we can expect *"Docker Swarm"* to make quick progress thanks to the new *"swarm mode"*

The Big 3 - Google Trends (2017)



Clearly Kubernetes is extending its lead in *Google "search trends"*

This is unsurprising given several recent announcements:

- AWS joined CNCF
- Docker announced they will support Swarm and Kubernetes
- Mesosphere updated their Kubernetes integration
- Google/VMWare/Pivotal agreement

Docker Swarm

Docker Swarm

Dec 2014 … **Docker Swarm** is announced

Orchestration using Docker Compose

Jun 2016 … **Swarm Toolkit** released

OpenSource Orchestration Toolkit

Jun 2016 … **Swarm Mode** announced

Orchestration integrated into Docker 1.12+ Engine

Oct 2017 … **Future Kubernetes support** announced

Orchestration integrated into Docker Engine

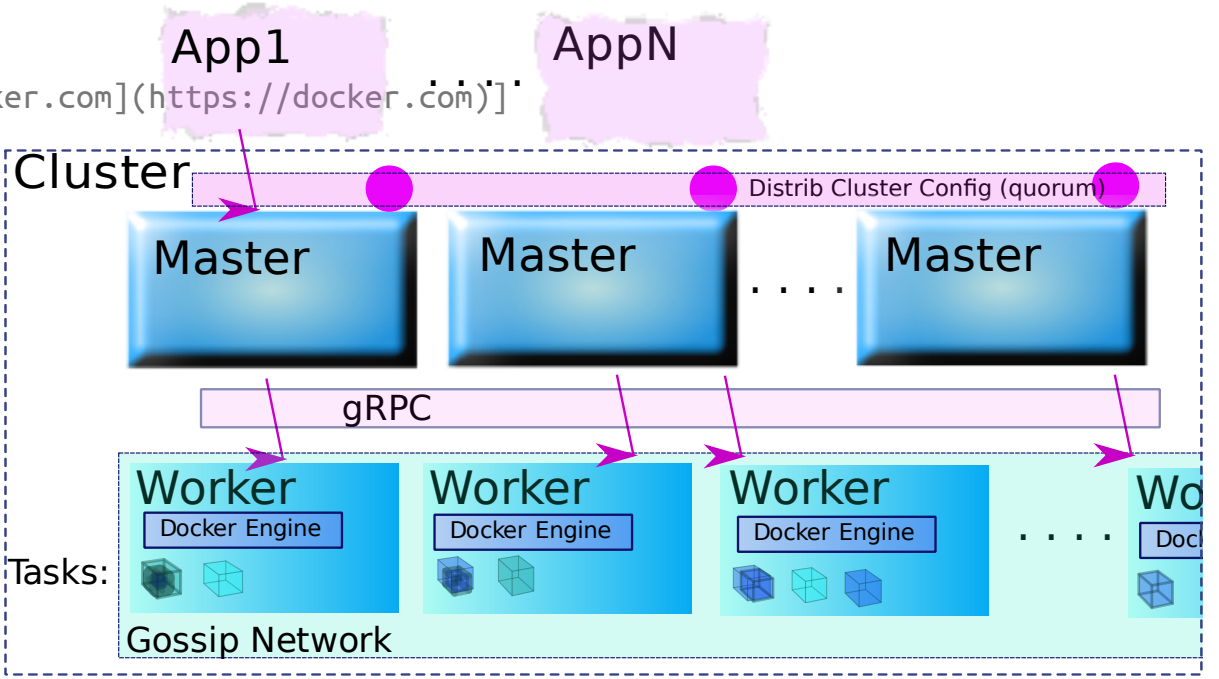
Swarm Mode brought:

- Orchestration directly in the Docker Engine
- Advanced networking features - mesh network, vxlan
- Load balancing, Service Discovery
- all with the traditional *Docker* ease of use

docker.com @mjbright

Architecture

```
## Docker Swarm
.footnote[.red[][docker.com](https://docker.com)]
```



Tasks are containers (today)
Stacks are groups of services

Docker Swarm

Networks

CNM: Container Networking Model (libnetwork)

NOTE: In future will provide CNI (CNCF/Kubernetes)

- null
- bridge
- overlay
- macvlan
- network plugins

Docker Swarm

Storage

- implicit volumes - no persistence
- implicit named volumes - no persistence
- explicit volumes - persistence
- explicit volumes - persistence

Docker Swarm

Using Docker "Swarm Mode"

Create a new swarm by creating the master node:

```
$ docker swarm init --advertise-addr 192.168.2.100
Swarm initialized: current node (dxn1zf6l61qsb1josjja83ng
```

Join a new Worker node to the swarm:

```
$ docker swarm join --token TOKEN 192.168.2.100:2377
```

Join a new Master node to the swarm:

```
$ docker swarm join-token manager
```

Docker Swarm Demo

Docker Swarm

Docker Swarm Demo

- Creation of a 3 node cluster
- Run a service on the cluster and scale it to 3 replicas
- Make a rolling update of the service
- Rollback
- Drain a node of the cluster
- Stack Creation

Docker Swarm - Hands-On

Tools:

- play-with-docker.com "PWD"
- katacoda.com
- docker-machine - tutorials, used locally or with cloud providers

Tutorials:

- [training.play-with-docker.com]

Kubernetes

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

Kubernetes

Google created based on extensive experience running containers internally ~ billions of containers a year

Started Oct 2014, reached v1.0 in July 2015, now at v1.8

Managed by the Cloud Native Computing Foundation CNCF: https://cncf.io/

Commercial offerings from CoreOS (Tectonic) and Canonical

Adopted by many others, integrated into their platforms.

kubernetes.io

Kubernetes

Integrated into:

- GKE (Google Container Engine)
- Microsoft Azure (Azure Container Service)
- VMWare/Pivotal/Google
- OpenStack above Kubernetes
  - Stackanetes (CoreOS, uses Tectonic)
    - Self healing OpenStack demo
  - Mirantis, Oracle (OpenStack CI/CD based on Kolla)

Various PaaS:

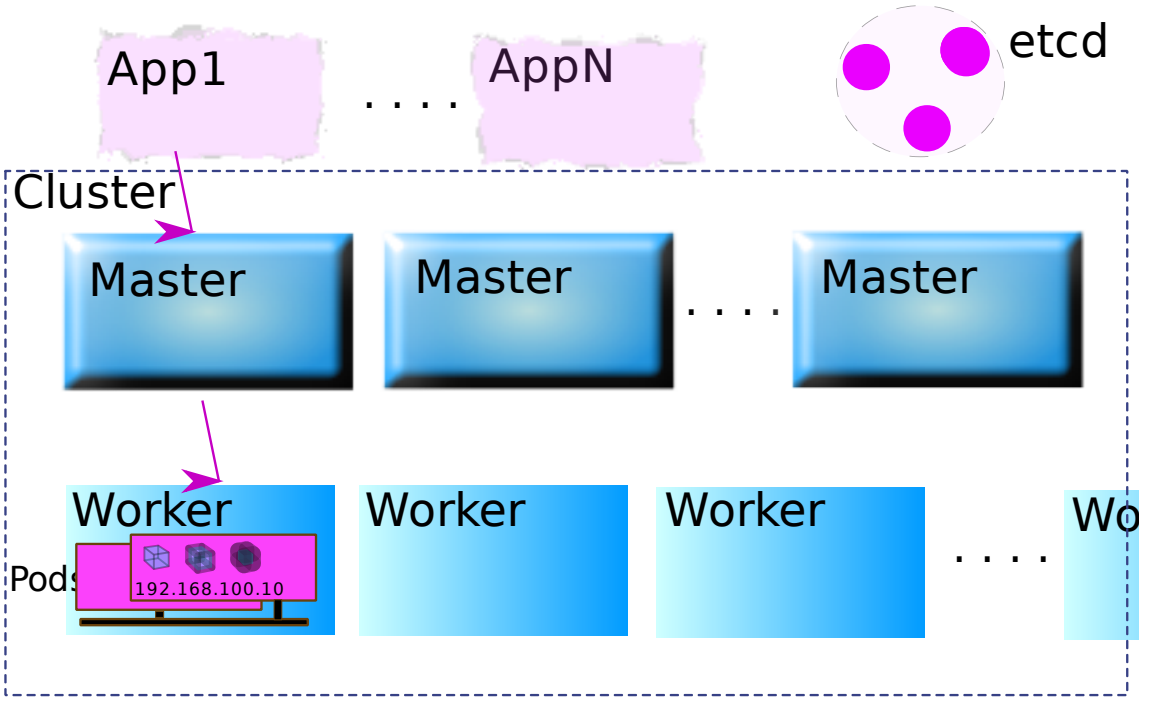
- RedHat OpenShift CP
- µ-Focus Stackato v4.0
- Deis (now Microsoft)

Many tools and add-ons available

- Dashboard
- Prometheus (monitoring)
- Heapster (monitoring)
- Helm (package manager)
- kubeadm, kops, ...

Architecture

Kubernetes

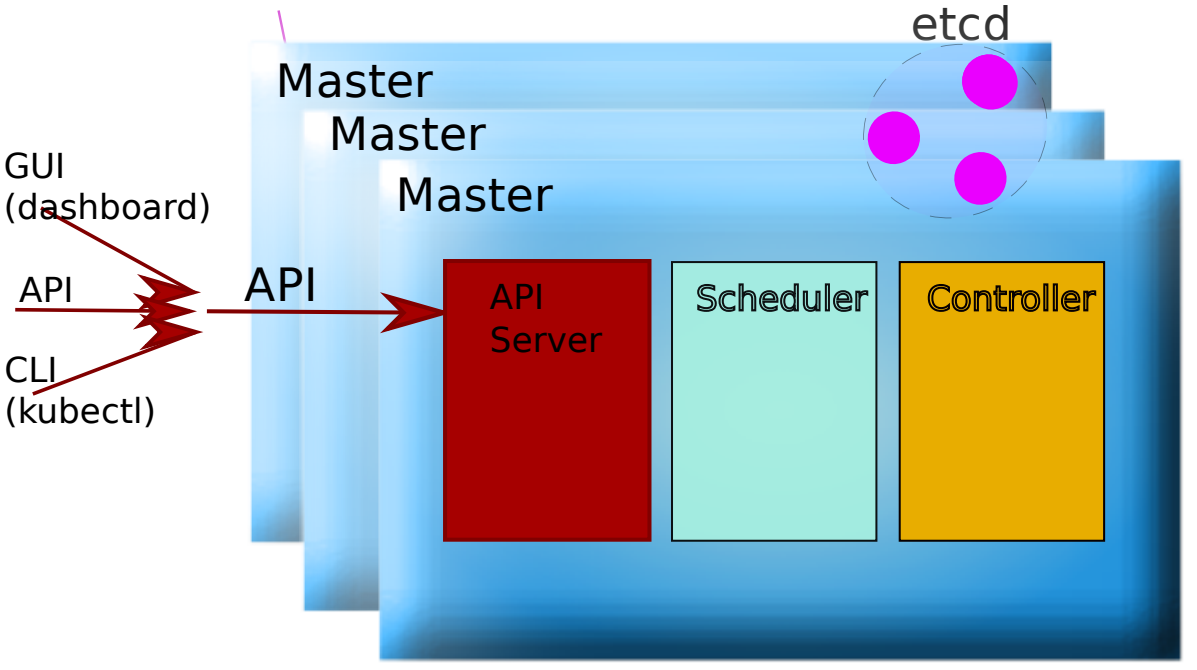


kubernetes.io @mibright

Deployments, Services (name, LB), Pods, Containers, ReplicaSets, DaemonSet, CronJob

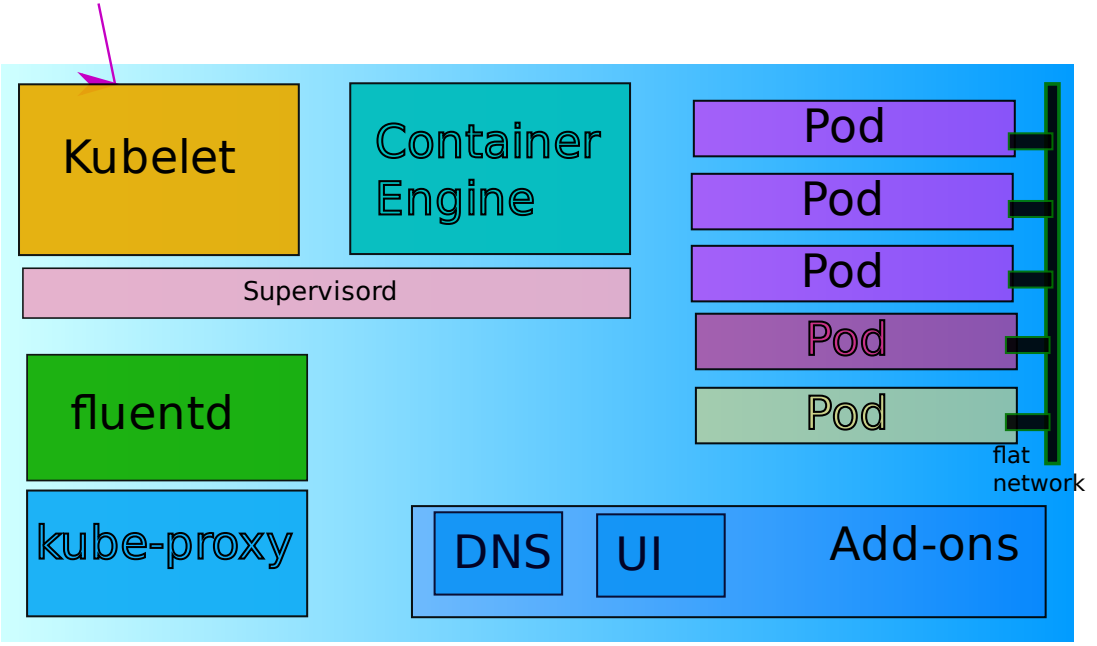
Architecture - Master Nodes

Kubernetes



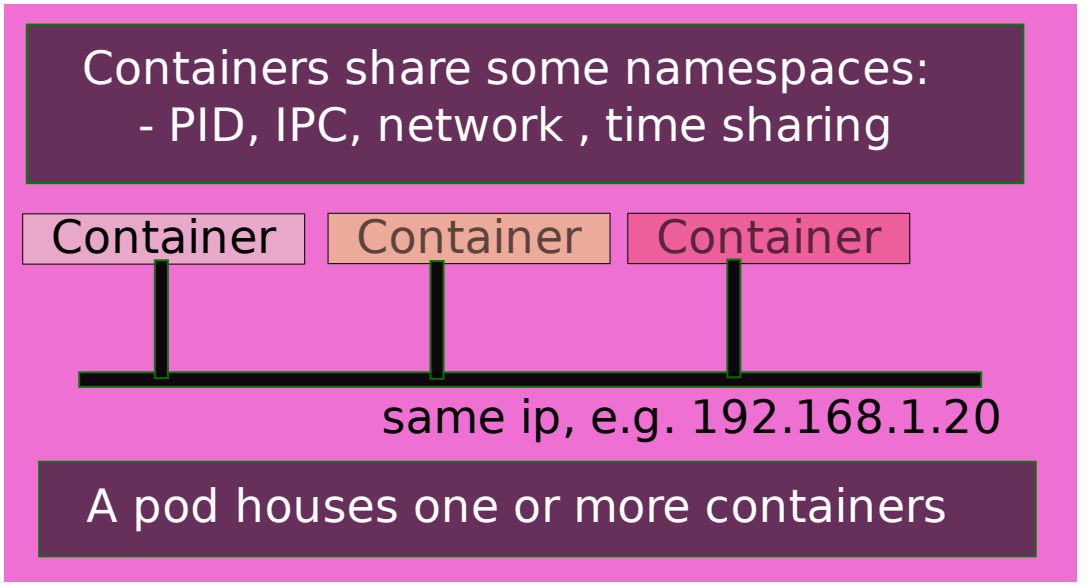
Architecture - Worker Nodes

Kubernetes



Architecture - Pods

Kubernetes



Kubernetes

Kubernetes Concepts

- Pod
  - smallest unit: 1 or more containers on the same IP
- ReplicaSet
  - Ensures that a given number of pods run
- Deployment
  - Declaratively manage ReplicaSets, rolling upgrade
- Service
  - addressable IP, load-balancing
- Labels
  - Key/value pairs which can be attached to objects
- Selectors
  - Used to select a group of objects by label
- Secrets
  - Small amounts of sensitive info (certs., keys, ..)
- ConfigMaps
  - Dynamically loadable config (k-v pairs, or json)
- DaemonSets
  - Ensure a pod runs on all (or subset of) nodes
- Jobs
  - Short-lived pods, e.g. for batch jobs

kubernetes.io

Kubernetes Demo

Kubernetes

Demo

- Creation of a Pod
- Scaling of Pod
- Exposing as a Service
- Make a rolling update of the service
- Rollback
- Drain a node of the cluster
- Stack Creation

Kubernetes - Hands-On

Tools:

- katacoda.com
- play-with-kubernetes.com "PWK"
- minikube - Single node cluster: tutorials, used locally with several hypervisors
- minishift - Single node OpenShift cluster: similar to Minikube+

Tutorials:

- Basics - http://kubernetes.io/docs/tutorials/kubernetes-basics/
- GCP - https://github.com/kubernetes/kubernetes/tree/master/examples

Apache Mesos

Apache Mesos

The most proven orchestrator today, exists since 2009.
Can scale to ~ 10,000 nodes.

Mesos is used in production by:

- Uber
- Twitter
- Paypal
- Hubspot
- Airbnb
- eBay
- Groupon
- Netflix

Mesos has Containerizers (Mesos, Docker) to isolate tasks.

Unified Containerizer supports Docker and appc image formats.

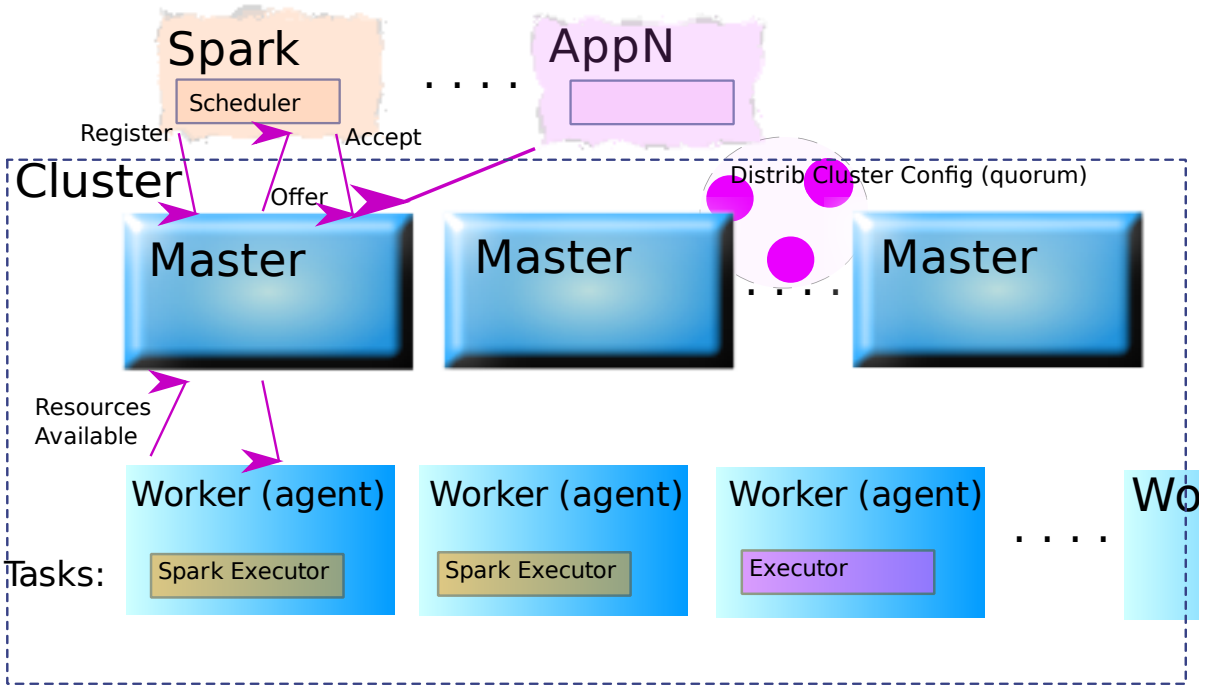
MesoSphere sells the **DC/OS** product based on **Mesos**.
DC/OS is also an open source project.

mesos.apache.org

MESOS

Apache Mesos

Architecture



A Mesos application provides its own scheduler and a task executor.

mesos.apache.org

Apache Mesos

Mesos is used in conjunction with Frameworks such as

- For *long running* tasks:
  - **Marathon** (Mesosphere), Aurora or Singularity
- For job orchestration:
  - Chronos "cron", Jenkins
- For Big Data Processing:
  - Hadoop, Spark, Storm
  - Cassandra, ElasticSearch, ...

mesos.apache.org @mjbright

Apache Mesos Demo

Mesos

Demo

- TODO ...
- Stack Creation

Apache Mesos - Hands-On

Tools:

- katacoda.com
- minimesos - tutorials, from Container Solutions
- DC/OS - From Mesosphere

Tutorials:

- Basics - TODO

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

... let's just compare them ...

What's common

Docker Swarm, Kubernetes, Apache Mesos are all creating rich Orchestration stacks with integrated container runtimes.

They're moving incredibly quicky ...

They are adding features such as networking capabilities, load balancing, services, labels.

They have a more 'declarative' approach allowing to abstract the data center as a single compute resource.

They support or are looking to support different runtime engines (*)

What advantages?

Docker "Swarm Mode"

Docker focus

- Security
- Portability (OS, Orch, Archi)
- Ease of use

Simple to use (despite underlying complexity)

All-in-one container engine plus orchestration

Uses Docker API and familiar docker commands

Advanced networking

```
- mesh networking
- Load Balancing and Service Discovery
```

Replication

Docker-EE (*Enterprise Edition*) commercial product

What advantages?

Kubernetes

Rich conceptual model, based on Google experience

Pods as groupings of containers

Labels and Selectors (for all components)

Networking - Load Balancing and Service Discovery

Replication

Large ecosystem with several commercial offerings:

- Core OS Tectonic, Canonical
- PaaS (OpenShift Container Platform, Stackato)

and Public Cloud Support

- Google GKE, Azure ACS

What advantages?

Apache Mesos

Most mature

Battle tested by many service providers

Scales to 10,000 nodes

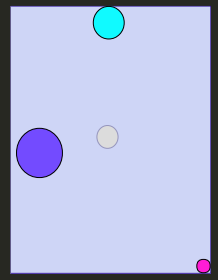
DataCenter OS - appears as 1 resource

Not just containers

Many frameworks available

(*Steeper learning curve*)

Thank you - Questions?



Latest slides available at:

https://containerorchestration.github.io/ContainerOrchestration/

@mjbright

Resources

@mjbright

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 2017 - Container Orchestration Wars, Karl Isenberg, Mesosphere
- 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

Kubernetes

Documentation - 1

- **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

Kubernetes

Documentation - 3

- **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.