

Introduction to Containers

Docker and Kubernetes

NA Partner Channels- IBM Cloud
IBM Cloud Technical Evangelist Team
January 24, 2018

Agenda

- 1 • Containers
- 2 • Why Containers?
- 3 • Docker Containers
- 4 • Container Orchestration
- 5 • Kubernetes
- 6 • Kubernetes in the IBM Cloud
- 7 • Conclusion



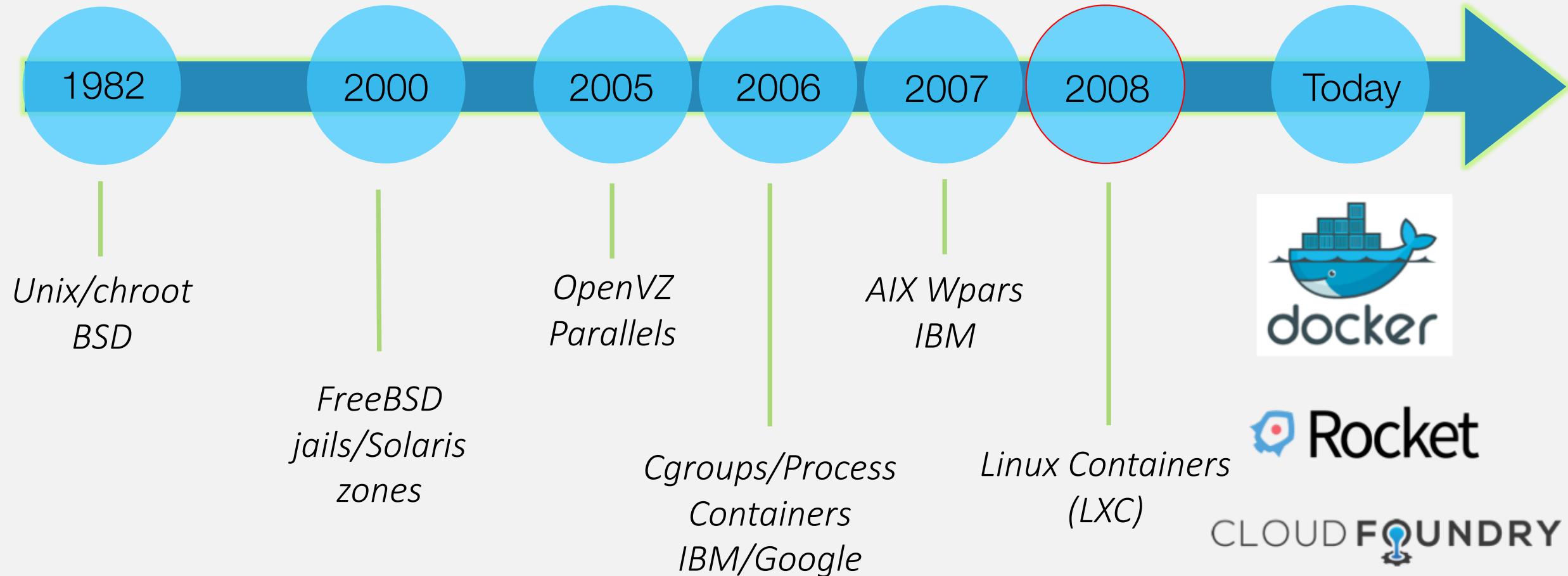
Why Containers?

Everyone Loves Containers

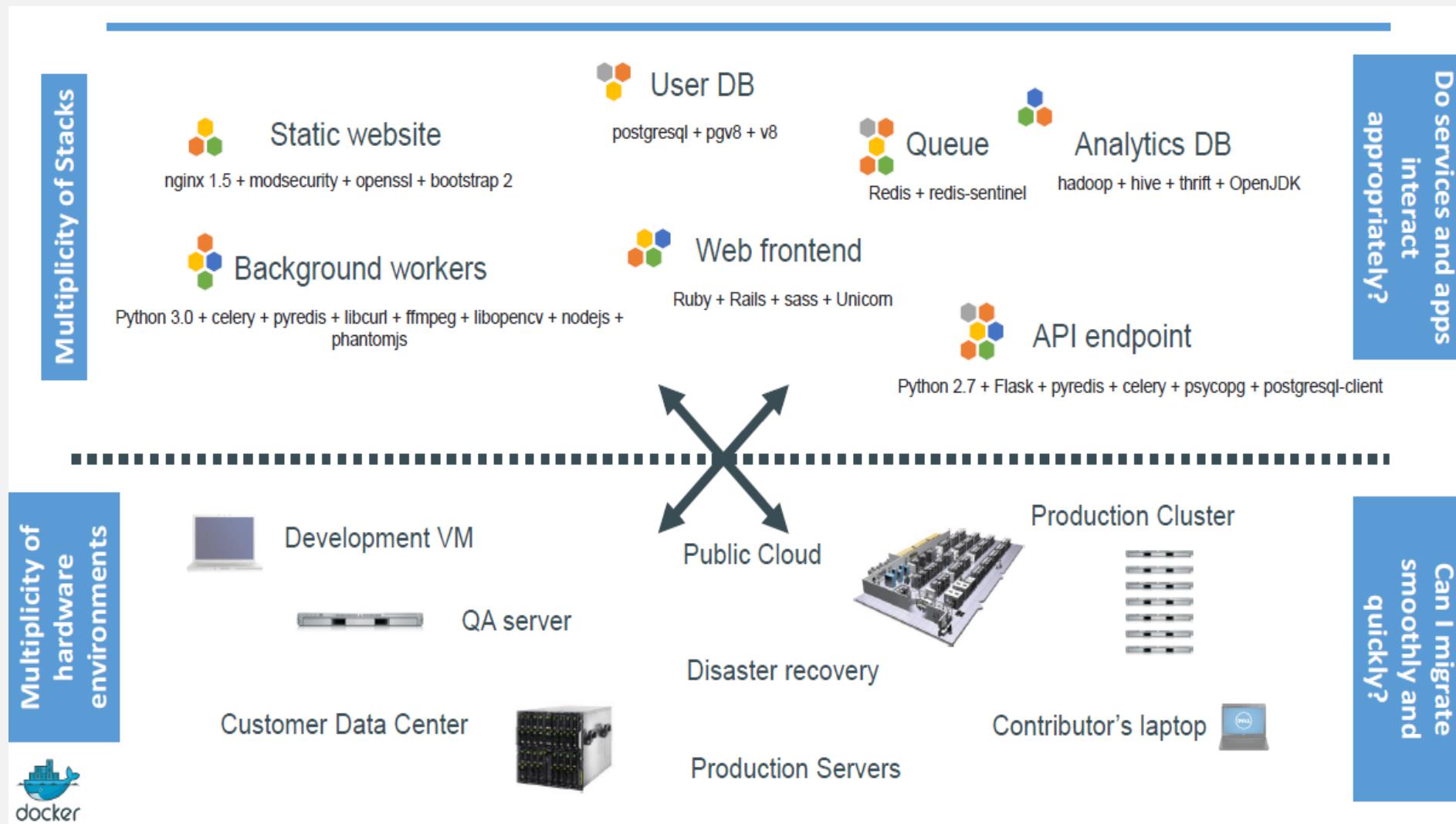


- A standard way to package an application and all its dependencies so that it can be moved between environments and run without changes
- Containers work by isolating the differences between applications inside the container so that everything outside the container can be standardized

Container History Lesson



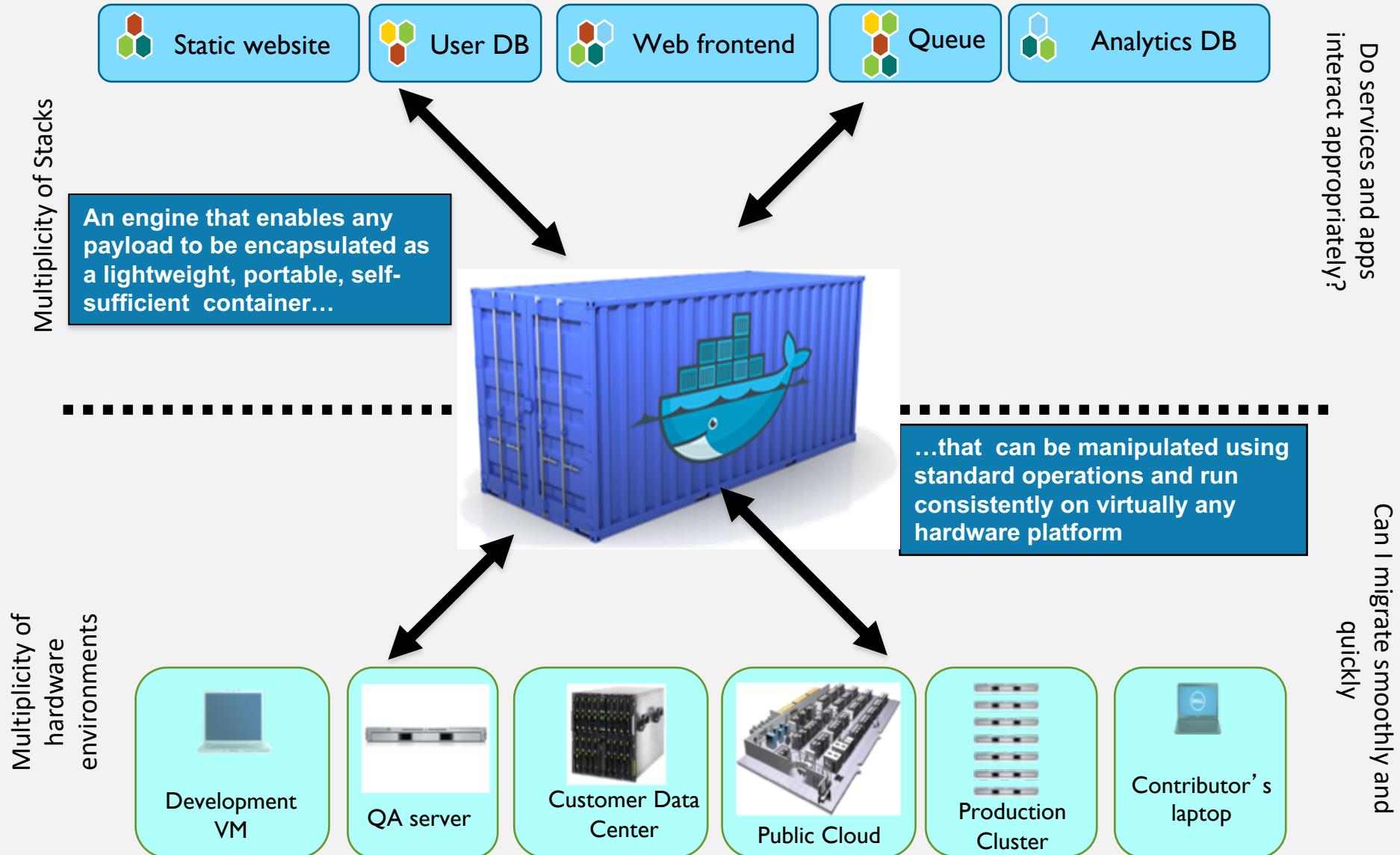
The Challenge



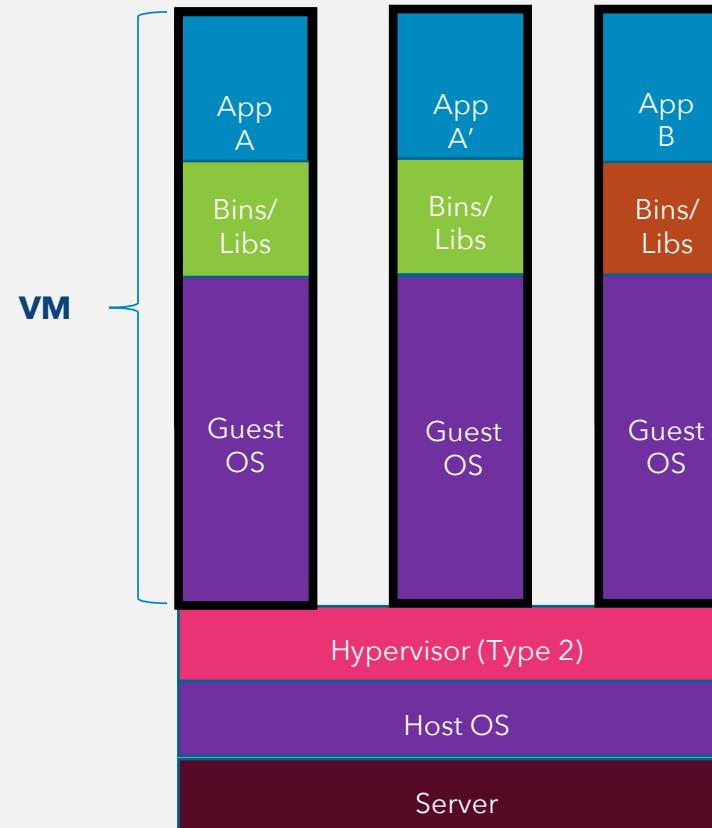
The Matrix from Hell

	Static website	?	?	?	?	?	?	?
	Web frontend	?	?	?	?	?	?	?
	Background workers	?	?	?	?	?	?	?
	User DB	?	?	?	?	?	?	?
	Analytics DB	?	?	?	?	?	?	?
	Queue	?	?	?	?	?	?	?
	Development VM	QA Server	Single Prod Server	Onsite Cluster	Public Cloud	Contributor's laptop	Customer Servers	
								

A Shipping Container for Code

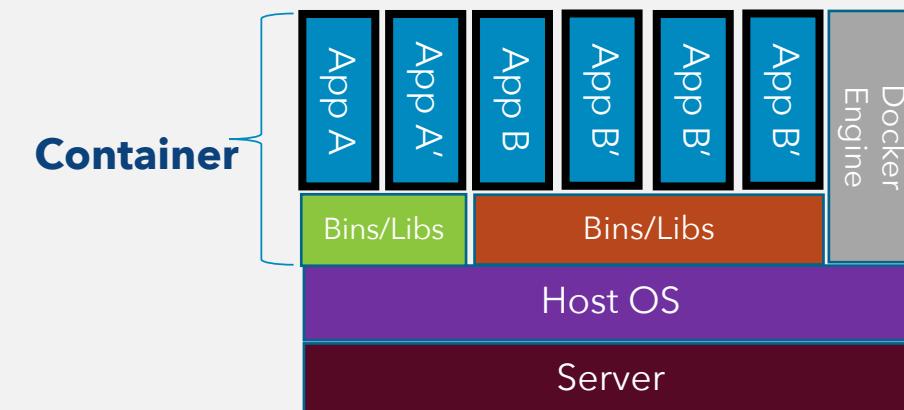


VM's vs. Containers



Containers are isolated, but share OS and, where appropriate, bins/libraries

...faster, less overhead



Dev vs. Ops

Dev

Ops



- **Code**
- **Libraries**
- **Configuration**
- **Server runtime**
- **OS**

- **Logging**
- **Remote access**
- **Network configuration**
- **Monitoring**

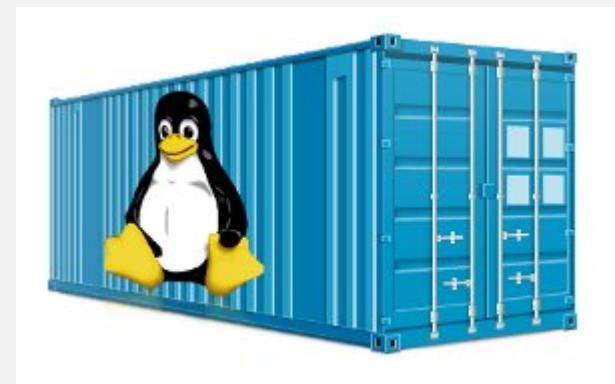
Separation of concerns

A container separates and bridges the Dev and Ops in DevOps

- Dev focuses on the application environment
- Ops focuses on the deployment environment

Container Advantages

- Containers are portable
- Containers are easy to manage
- Containers provide “just enough” isolation
- Containers use hardware more efficiently
- Containers are immutable



Docker Containers

Docker Adoption

Docker enables application development **efficiency**, making deployment more **efficient**, eliminating vendor 'lock-in' with true **portability**



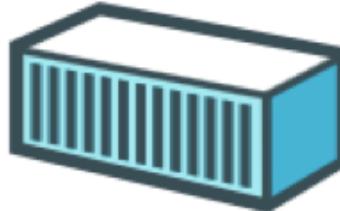
- Open software
- Open contribution
- Open design
- Open governance

Docker Mission

Docker is an **open platform** for building distributed applications for **developers** and **system administrators**



Build



Ship



Run



Any App

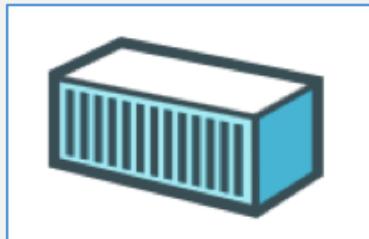
Anywhere

Docker Components



Image

- A read-only snapshot of a container stored in Docker Hub to be used as a template for building containers



Container

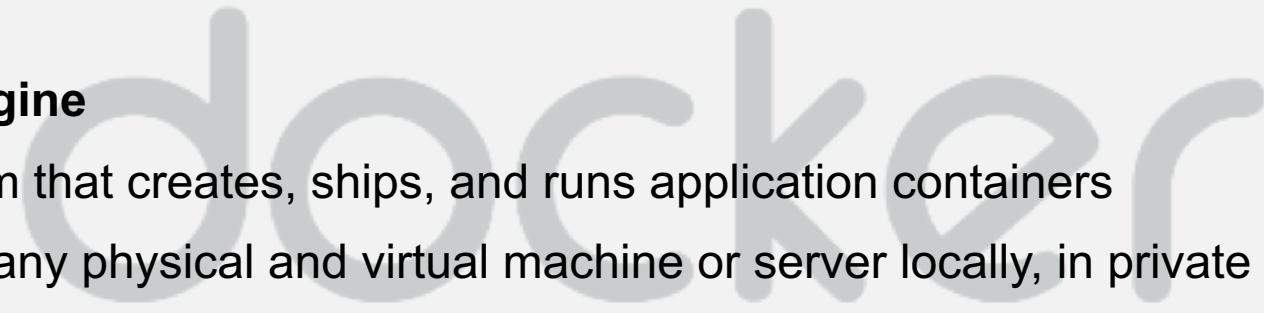
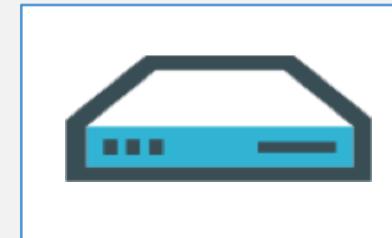
- The standard unit in which the application service resides or transported

Docker Hub/Registry

- Available in SaaS or Enterprise to deploy anywhere you choose
- Stores, distributes, and shares container images

Docker Engine

- A program that creates, ships, and runs application containers
- Runs on any physical and virtual machine or server locally, in private or public cloud
- Client communicates with Engine to execute commands



Containers



Everyone's container journey starts with one container....

Containers



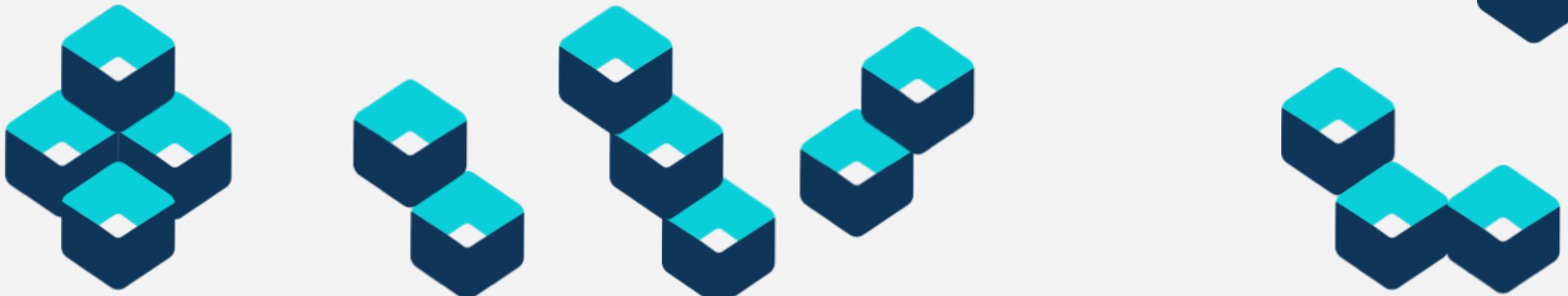
At first the growth is easy to handle....



Containers



But soon it is overwhelming... chaos reigns



Orchestration

More to Containers than just Docker

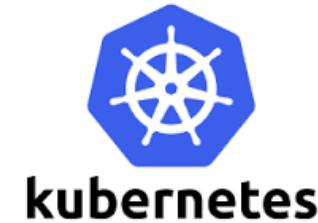
Serverless



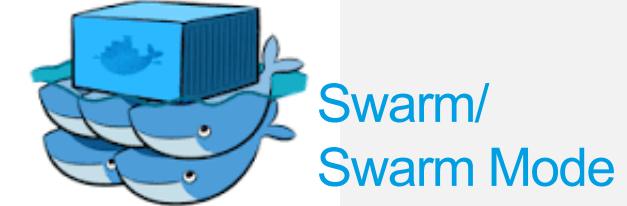
PaaS



Container Orchestration



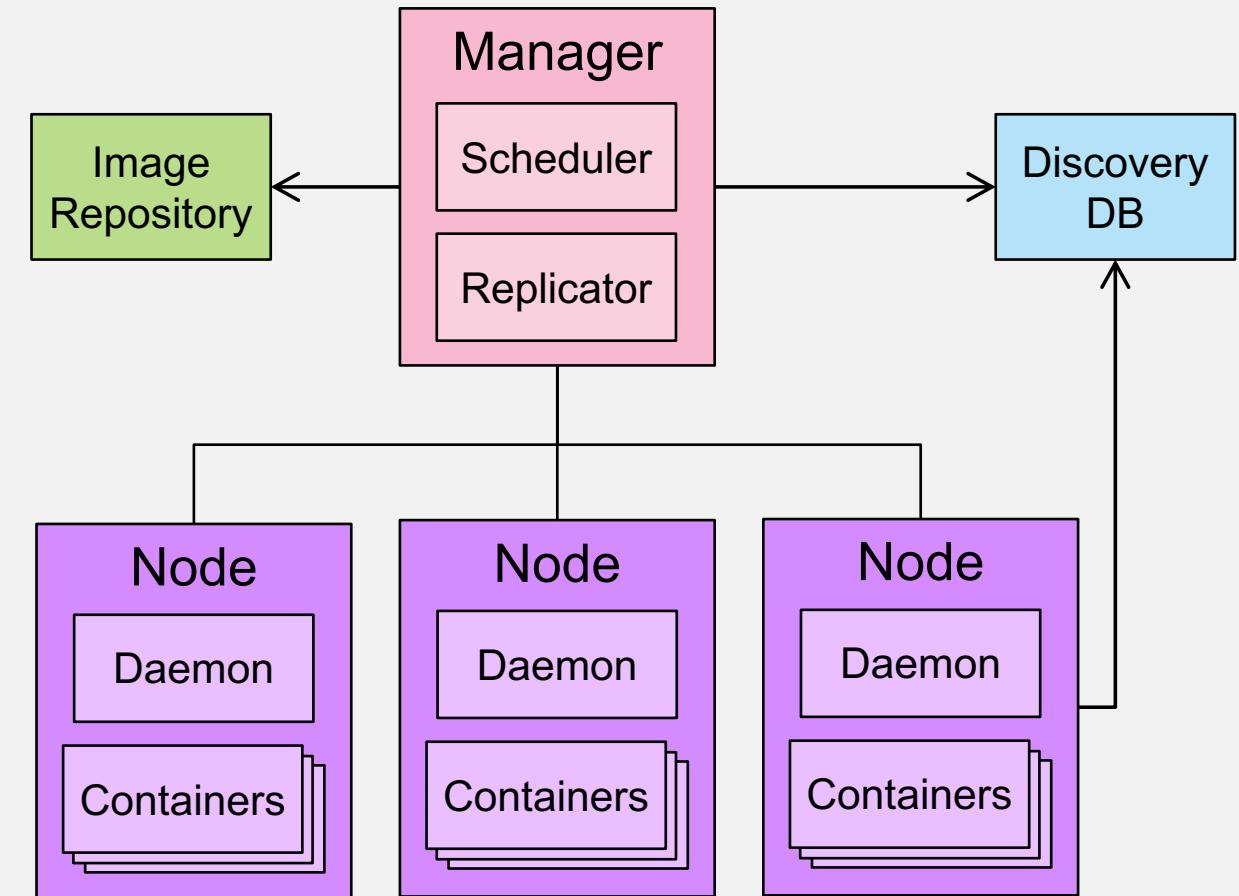
Container Engine



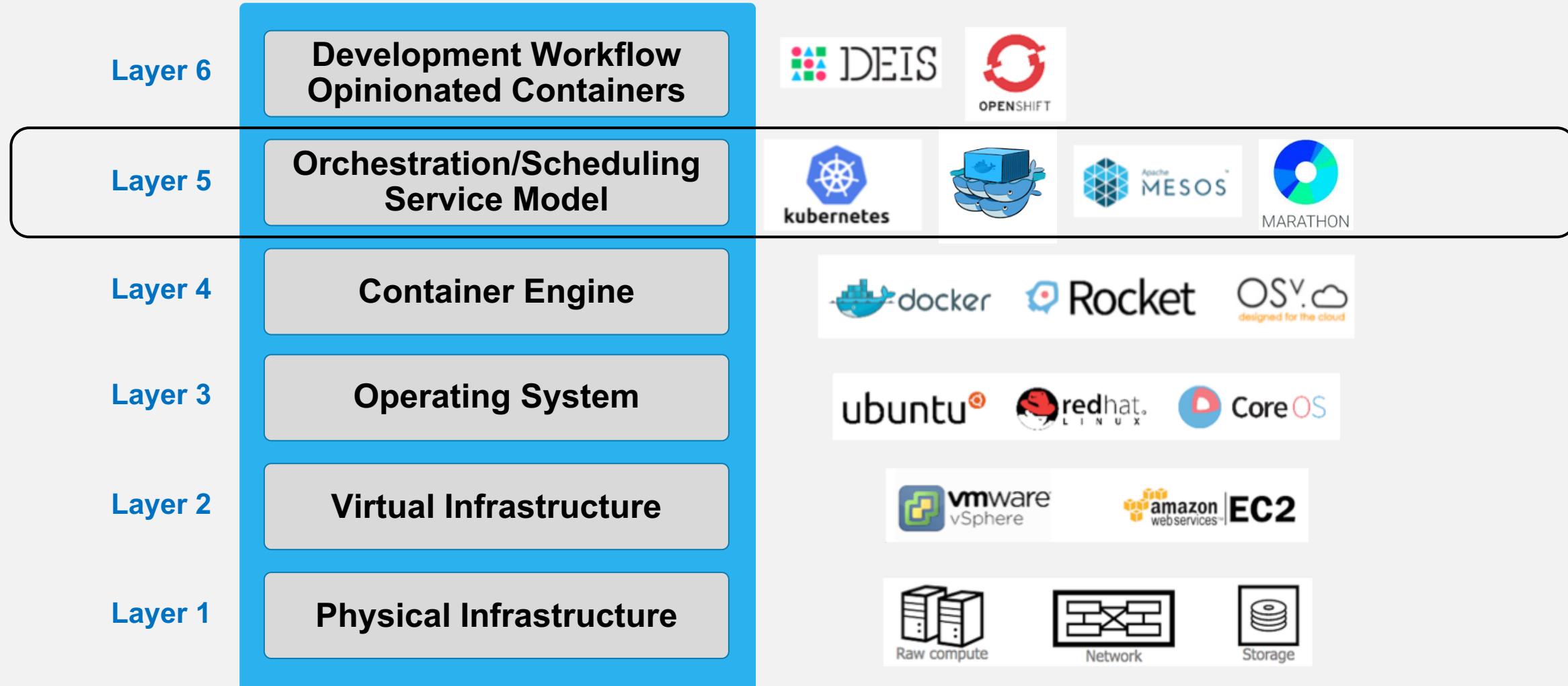
What is Container Orchestration?

- Container orchestration
 - Cluster management
 - Scheduling
 - Service discovery
 - Replication
 - Health management

Container Orchestrator



Container Ecosystem Layers

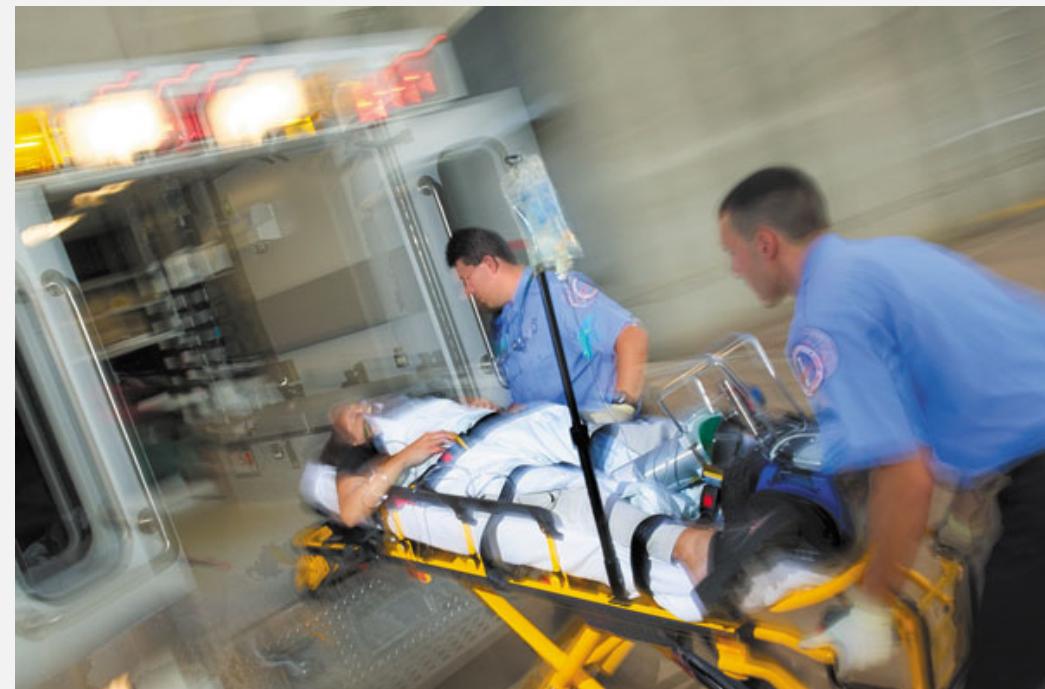


Kubernetes

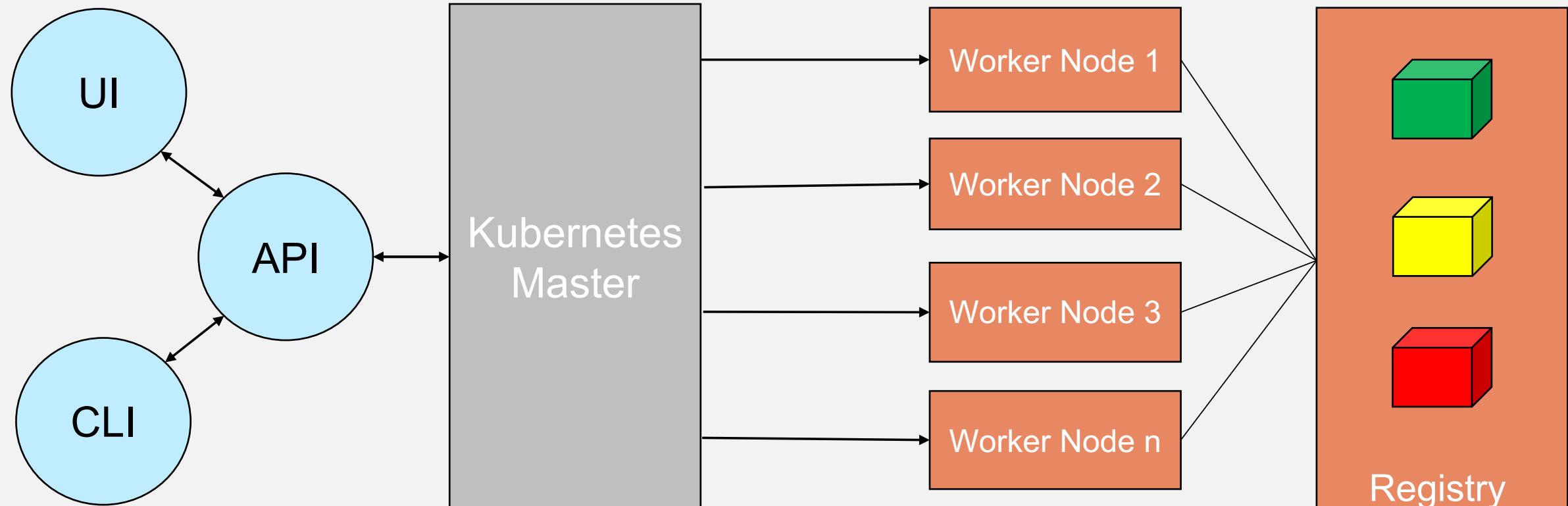


What is Kubernetes?

- Container orchestrator
- Manage applications, not machines
- Designed for extensibility
- Open source project managed by the Linux Foundation



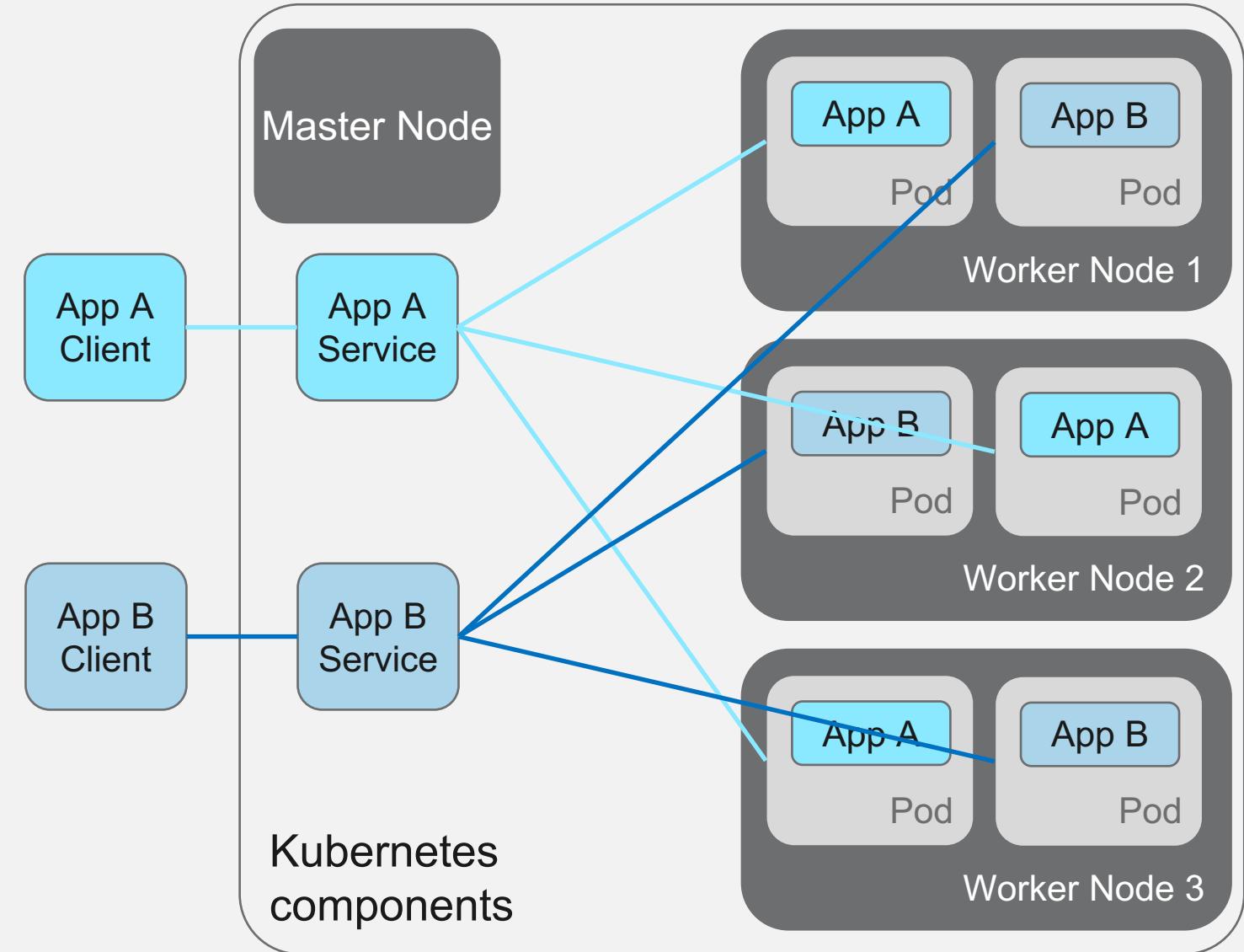
Kubernetes Architecture



- Etcd
- API Server
- Controller Manager Server
- Scheduler

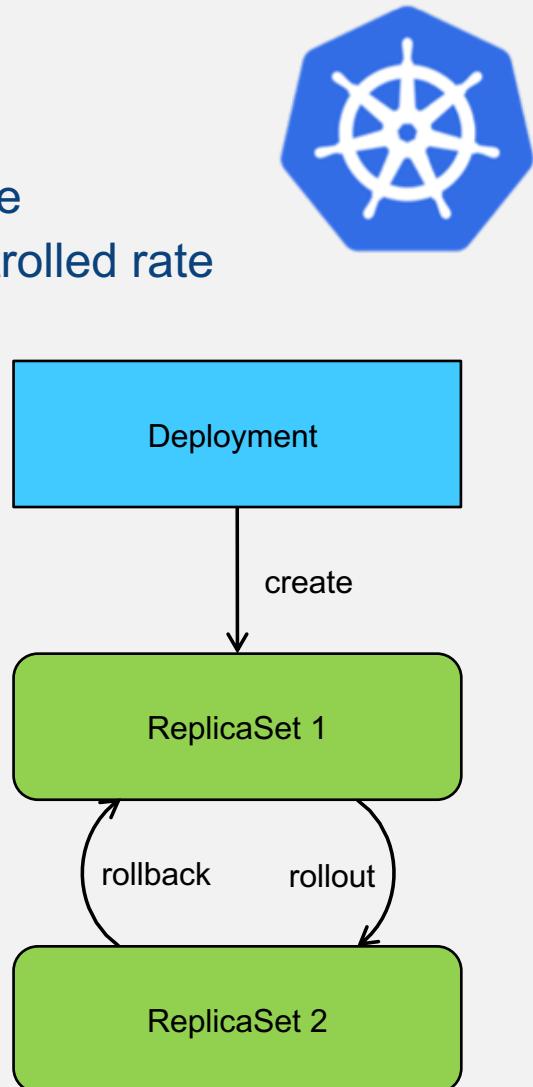
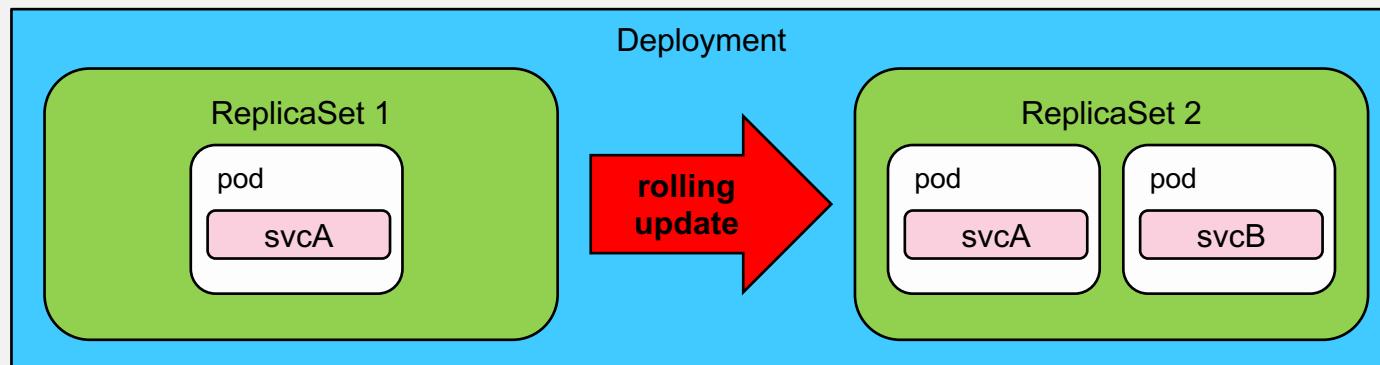
Kubernetes Architecture: Workloads

- Container
 - Packaging of an app
- Pod
 - Unit of deployment
- Service
 - Fixed endpoint for 1+ pods



Kubernetes Terminology: Deployment

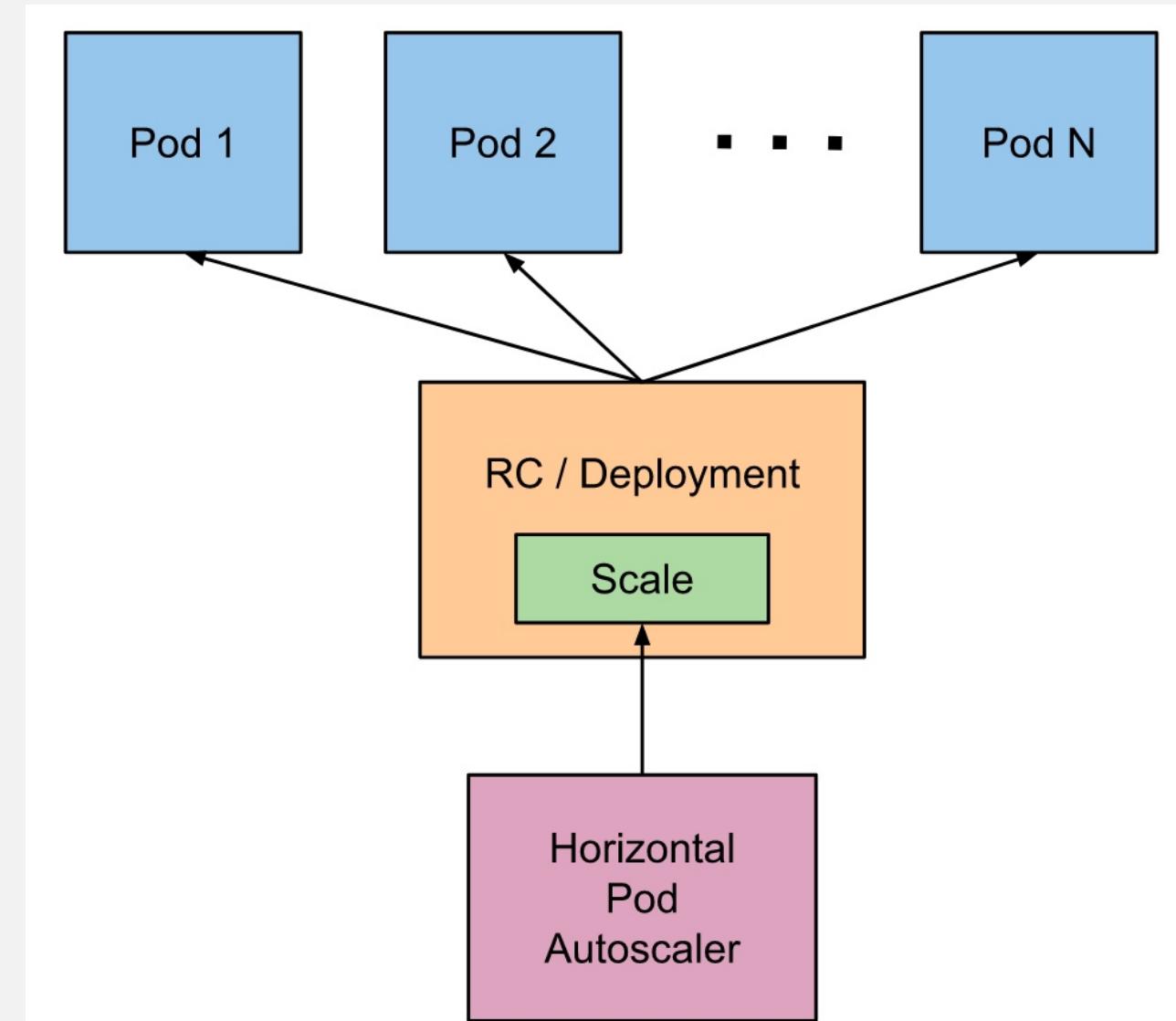
- Deployment
 - A set of pods to be deployed together, such as an application
 - Declarative: Revising a Deployment creates a ReplicaSet describing the desired state
 - Rollout: Deployment controller changes the actual state to the desired state at a controlled rate
 - Rollback: Each Deployment revision can be rolled back
 - Scale and autoscale: A Deployment can be scaled
- ReplicaSet
 - The next-generation ReplicationController
 - A set of pod templates that describe a set of pod replicas
 - Uses a template that describes specifically what each pod should contain
 - Ensures that a specified number of pod replicas are running at any given time



Kubernetes Terminology: Autoscaling

- Horizontal Pod Autoscaling (HPA)

```
$ kubectl autoscale deployment  
<deployment-name> --cpu-percent=50  
--min=1 --max=10 deployment  
" <hpa-name>" autoscaled
```



Common Kubernetes Commands

- Get the state of your cluster
`$ kubectl cluster-info`
- Get all the nodes of your cluster
`$ kubectl get nodes -o wide`
- Get info about the pods of your cluster
`$ kubectl get pods -o wide`
- Get info about the replication controllers of your cluster
`$ kubectl get rc -o wide`
- Get info about the services of your cluster
`$ kubectl get services`
- Get full config info about a Service
`$ kubectl get service NAME_OF_SERVICE -o json`
- Get the IP of a Pod
`$ kubectl get pod NAME_OF_POD --template={{.status.podIP}}`
- Delete a Pod
`$ kubectl delete pod NAME`
- Delete a Service
`$ kubectl delete service NAME_OF_SERVICE`

Helm & Helm Chart

- Helm
 - Package manager for Kubernetes
 - Used to manage Kubernetes applications
- Helm Chart
 - Used to define, install, and upgrade complex Kubernetes applications
 - Easy to create, version, share and publish
 - Expressed in “Yet Another Markup Language” (YAML) files



Kubernetes in the IBM Cloud

Kubernetes in the IBM Cloud

LEVERAGING KUBERNETES IN IBM CLOUD PUBLIC AND PRIVATE

- Combine the power of open source with the scale of the IBM Cloud
- Accelerate Innovation and improve time-to-market by empowering developers to rapidly build cloud-native and cloud compatible applications for both new and existing workloads
- Integrate cloud and on-prem applications to a host of enterprise-scale capabilities and services with the applications you are developing such as analytics, Machine Learning, AI (Watson), middleware, and IoT services.
- Leverage operational services to help you manage your environment, including monitoring, log management, and security

Kubernetes in the IBM Cloud

TWO PRIMARY OPTIONS

IBM Cloud (Public)

- IBM Kubernetes Container Service available as a fully managed service, as well as private Docker registry
- Pods and Containers can leverage other IBM Cloud services, such as Watson AI, the Watson Data Platform, and many others
- Pods and Containers can access your services either on-prem or from other cloud providers through secure means
- Full DevOps services available to help manage application development

IBM Cloud Private

- An operational Kubernetes-focused development and production version of IBM Cloud for deployment in your on-prem or cloud environment
- Enterprise-quality management, scalability, security, and resiliency features to support your Kubernetes Cluster deployment
- Built-in access to enterprise services such as analytics, middleware, data storage, and data science
- Access and integrate to your other on-prem and/or cloud services

IBM Cloud Kubernetes Options

IBM Cloud

Catalog

Docs

Support

Manage



[View All](#)

Create new cluster

Provision a cluster of hosts, called worker nodes, to deploy and manage highly-available apps.

Region

US East

[Docs](#)

[Terms](#)

Plan type

Free

New to Kubernetes? Create a cluster with 1 worker node to explore the capabilities.

Free

Pay-As-You-Go plan

Create a fully-customizable, production-ready cluster with your choice of hardware

Starting from \$0.19 hourly

Create Cluster

Cancel

Cluster details

Cluster Name

mycluster

Location

Kubernetes version

IBM Cloud Private Options

Community

Use Case

Create cloud-native applications in a non-production environment

Platform

- Kubernetes
- Core services (security, logging, monitoring, etc.)
- Catalog of containerized content

Freely available in Docker Hub

The Community Edition is limited to 1 Master Node, and is for non-production use.

Cloud Native

Use Case

Create cloud-native applications in a non-production environment

Platform

- Kubernetes
- Core services (security, logging, monitoring, etc.)
- Catalog of containerized content

Cloud Foundry (optional add-on)

IBM Enterprise Software

- Microservice Builder
- WebSphere Liberty
- IBM SDK for node.js
- Cloud Automation Manager

Enterprise

Use Case

- Modernize and optimize existing applications
- Open enterprise data centers

Platform

- Kubernetes
- Core services (security, logging, monitoring, etc.)
- Catalog of containerized content

Cloud Foundry (optional add-on)

IBM Enterprise Software

- Everything in Cloud Native, plus:
 - + WAS ND
 - + MQ Advanced
 - + API Connect Professional
 - + Db2 Direct Advanced (separate PN)
 - + UrbanCode Deploy (separate PN)

Conclusion

Conclusion

- Why containers?
- Docker containers
- Container orchestration
- Kubernetes
- Kubernetes in the IBM Cloud

Resources

- Docker tutorial
 - <https://docs.docker.com/get-started/>
- Kubernetes tutorial
 - <https://kubernetes.io/docs/tutorials/kubernetes-basics/>
- The Evolution of Linux Containers and Their Future
 - <https://dzone.com/articles/evolution-of-linux-containers-future>
- Introduction to container orchestration
 - <https://www.exoscale.ch/syslog/2016/07/26/container-orch/>
- TNS Research: The Present State of Container Orchestration
 - <https://thenewstack.io/tns-research-present-state-container-orchestration/>
- Large-scale cluster management at Google with Borg
 - <https://research.google.com/pubs/pub43438.html>

