

Intro to Kubernetes Workshop



MAY
21

**Intro to Containers and
Kubernetes**

by Event hosted by IBM &
Dallas Kubernetes Meetup

Richard Irving

Agenda

-  What is Docker?
 -  What is Kubernetes?
 -  Install Minikube
 -  Deploy some containers
 -  Ask some questions
-

The Objective

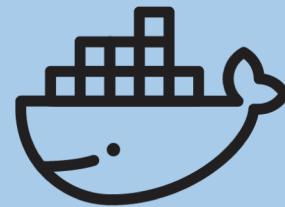
Get enough information and insight to begin experimenting with your own containerized workloads.

Where are the labs?

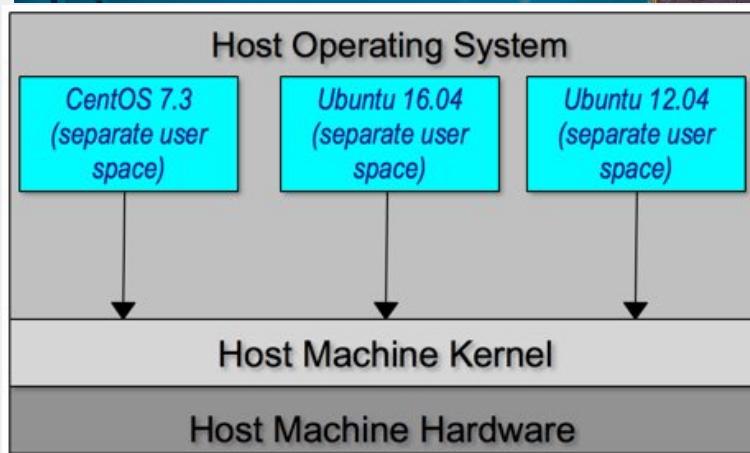


<https://github.com/irvnet/dallas-containerWorkshop-may2018>

What is Docker?



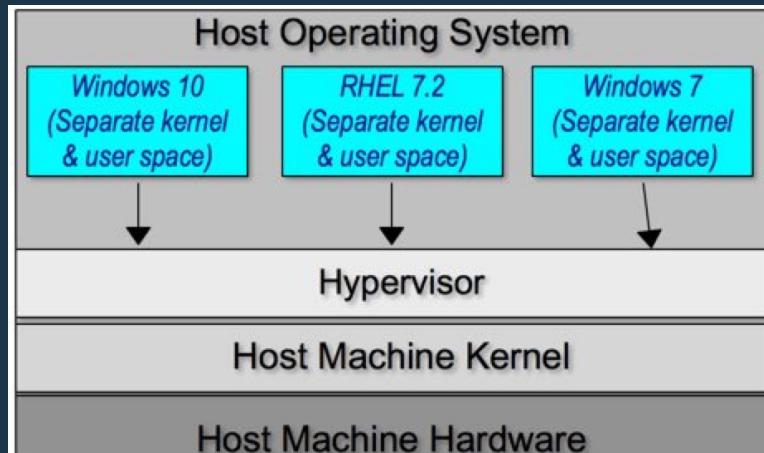
Containers



VMs

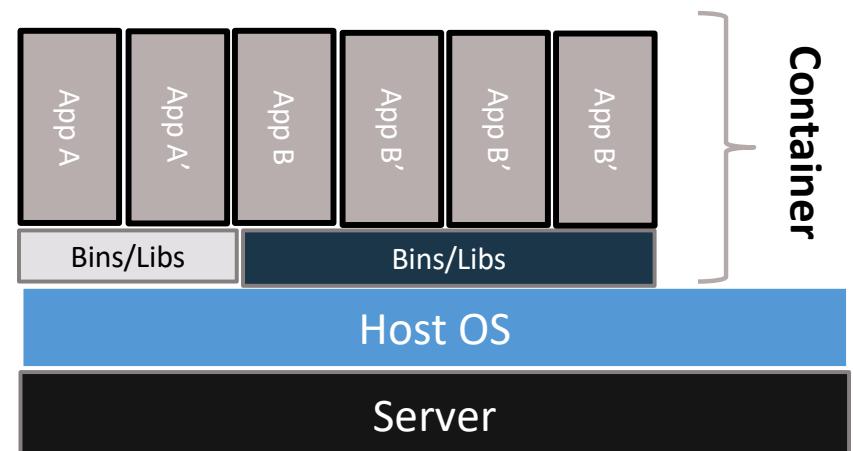


VS

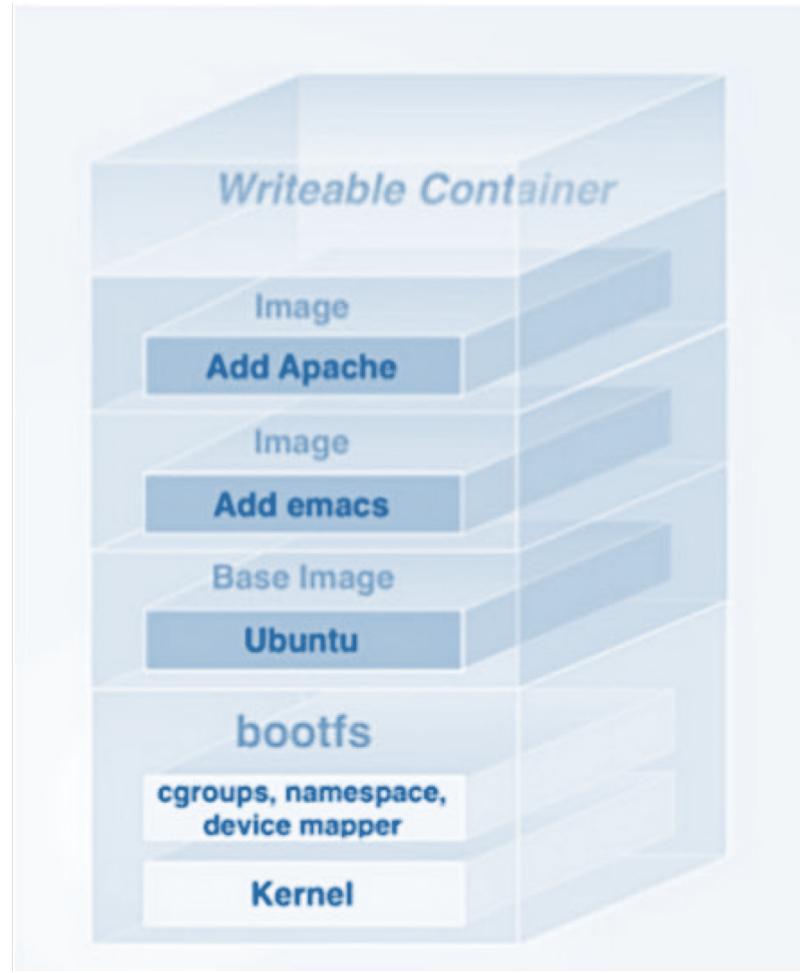


What is a container?

- Isolated userspace within a running linux OS
- Shared linux kernel across containers
- All packages and data in an isolated runtime saved as a filesystem
- Works on all the major linux platforms
- Looks like a vm from inside, like a normal process from outside
- Standardized packaging for applications and their dependencies that runs on any docker-enabled machine



Docker Image Structure



Separation of Concerns

Davidoff

The Developer

Handles what's "inside" the container

- His code
- His Libraries
- His Package Manager
- His Apps
- His Data

All Linux servers look the same

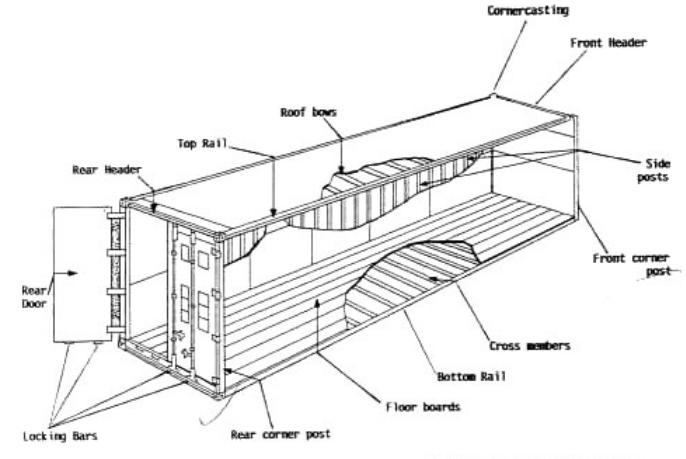
Oswald

The Ops Guy

Handles what's "outside" the container

- Logging
- Remote access
- Monitoring
- Network configuration

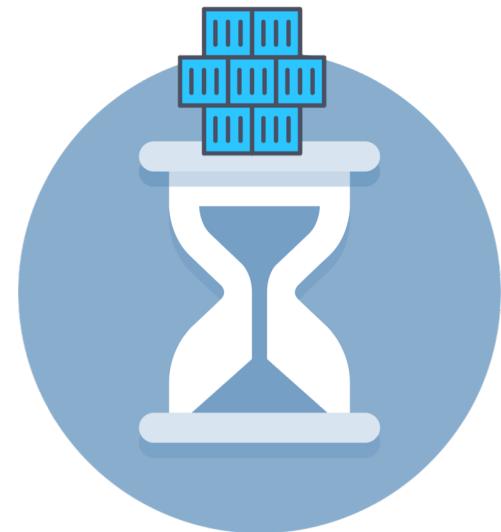
All containers start, stop, copy, attach, migrate, etc. the same way



Major components of the container:

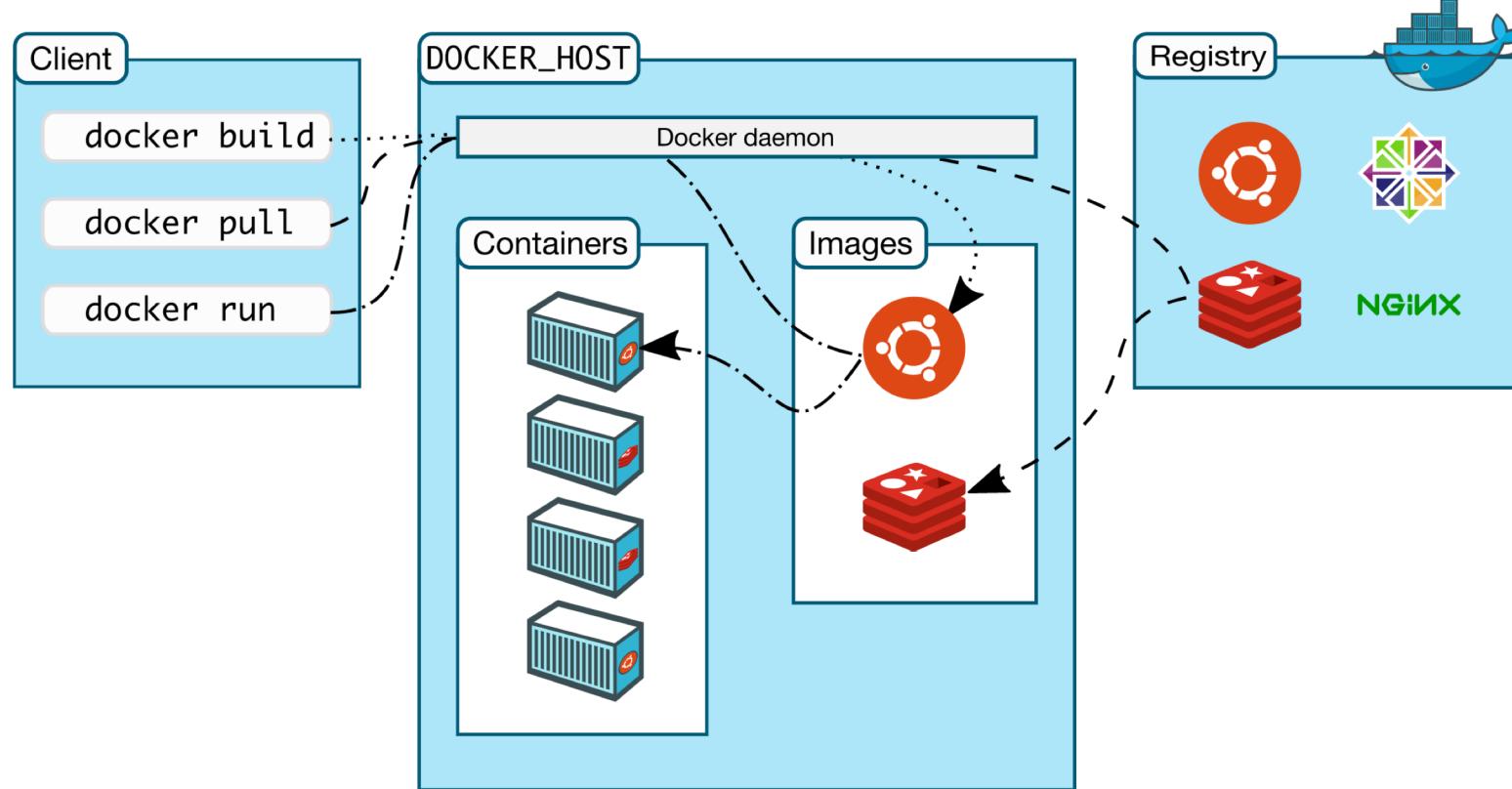
Mechanism	Operating system	License	Available since or between	File system isolation
chroot	Most UNIX-like operating systems	Varies by operating system	1982	Partial ^[a]
Docker	Linux, ^[7] FreeBSD, ^[8] Windows x64 (Pro, Enterprise and Education) ^[9] macOS ^[10]	Apache License 2.0	2013	Yes
Linux-VServer (security context)	Linux, Windows Server 2016	GNU GPLv2	2001	Yes
lmbtify	Linux	Apache License 2.0	2013	Yes
LXC	Linux	GNU GPLv2	2008	Yes ^[12]

Containers weren't born yesterday...

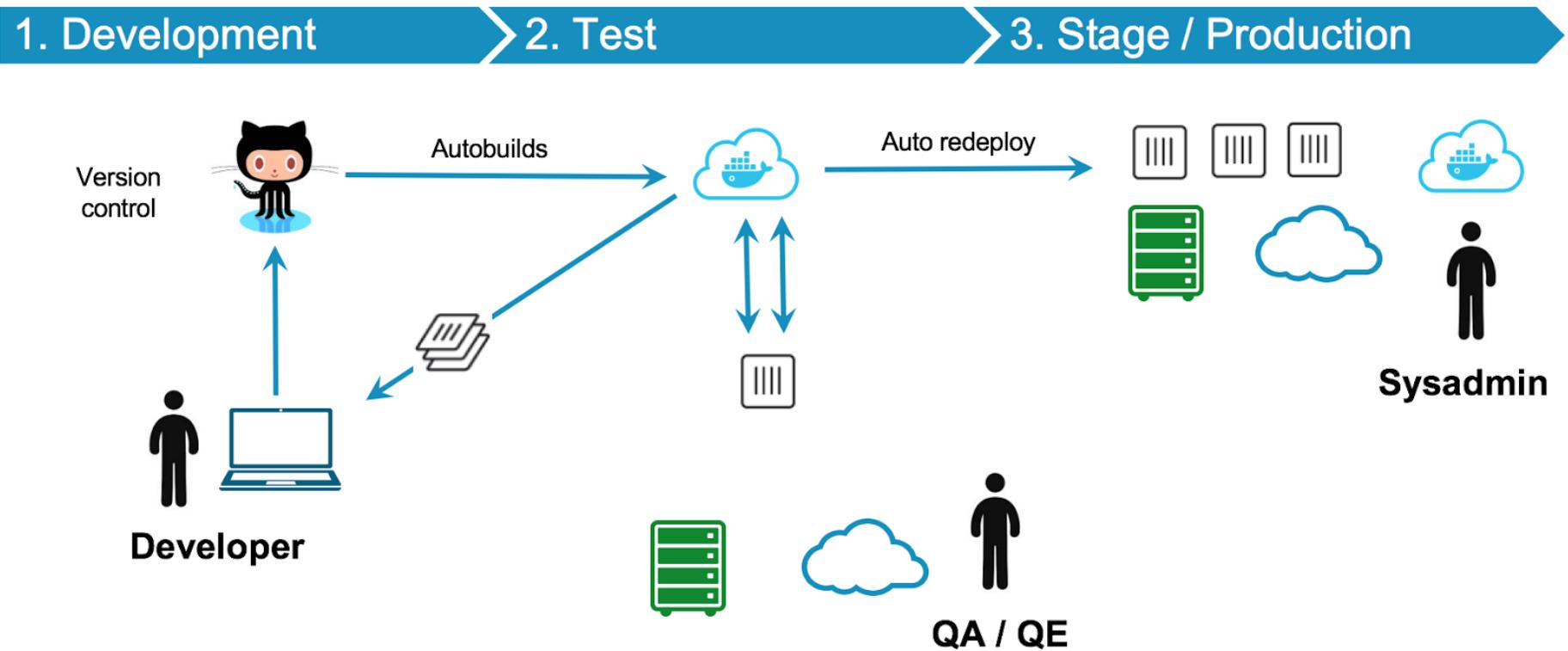


https://en.wikipedia.org/wiki/Operating-system-level_virtualization#Implementations

Docker Architecture



Docker Workflow



What is Kubernetes?



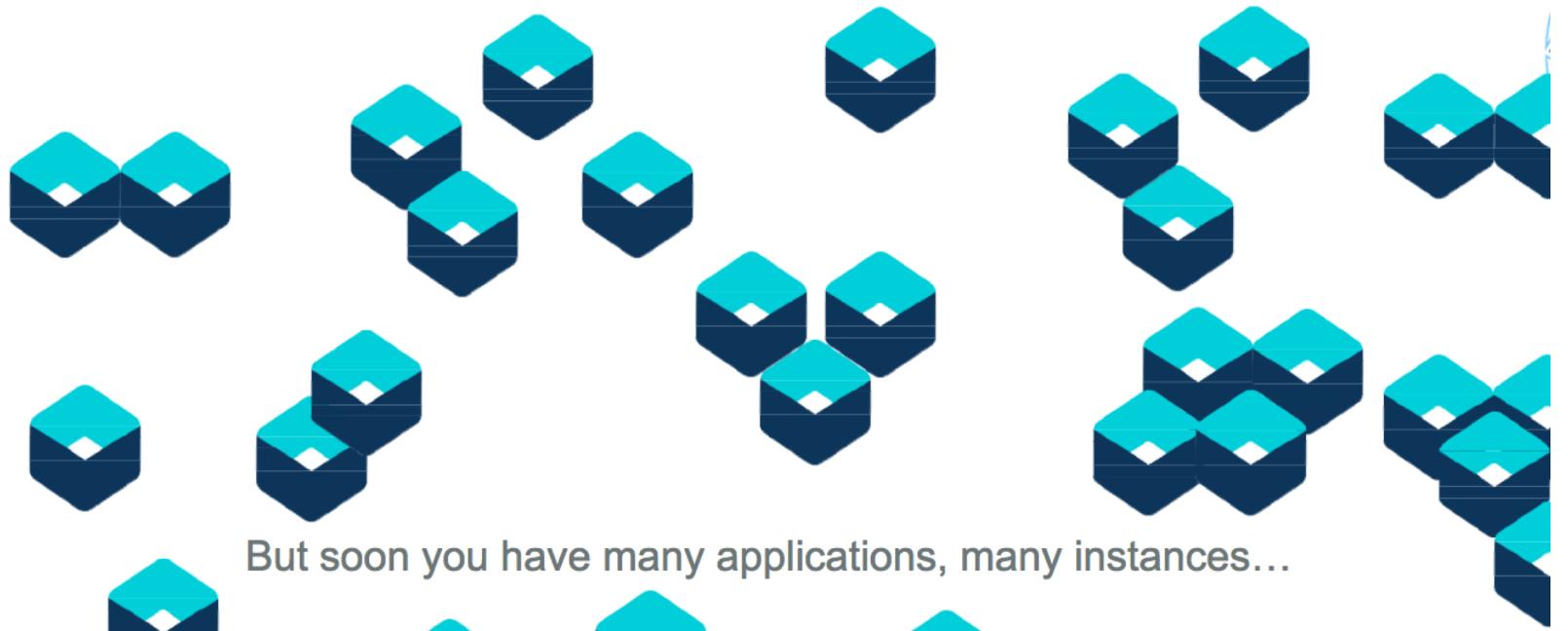


Everyone's container journey starts with one container....

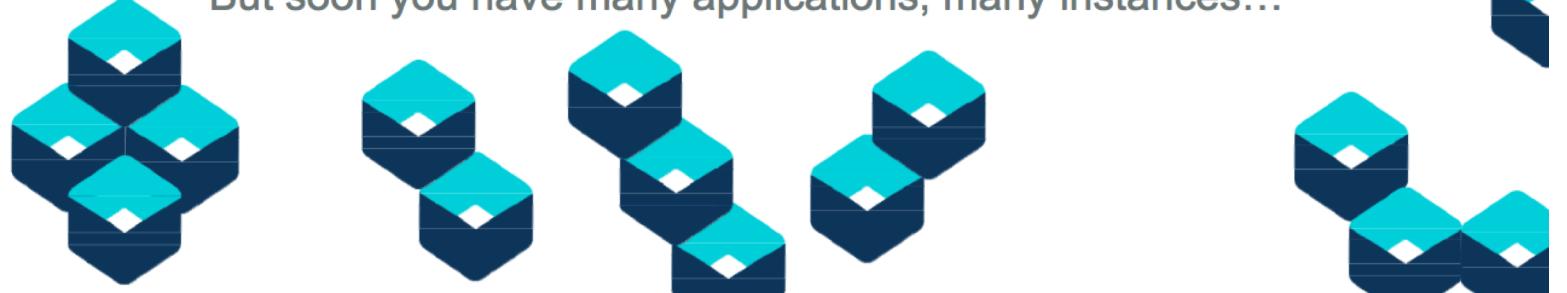


At first the growth is easy to handle....

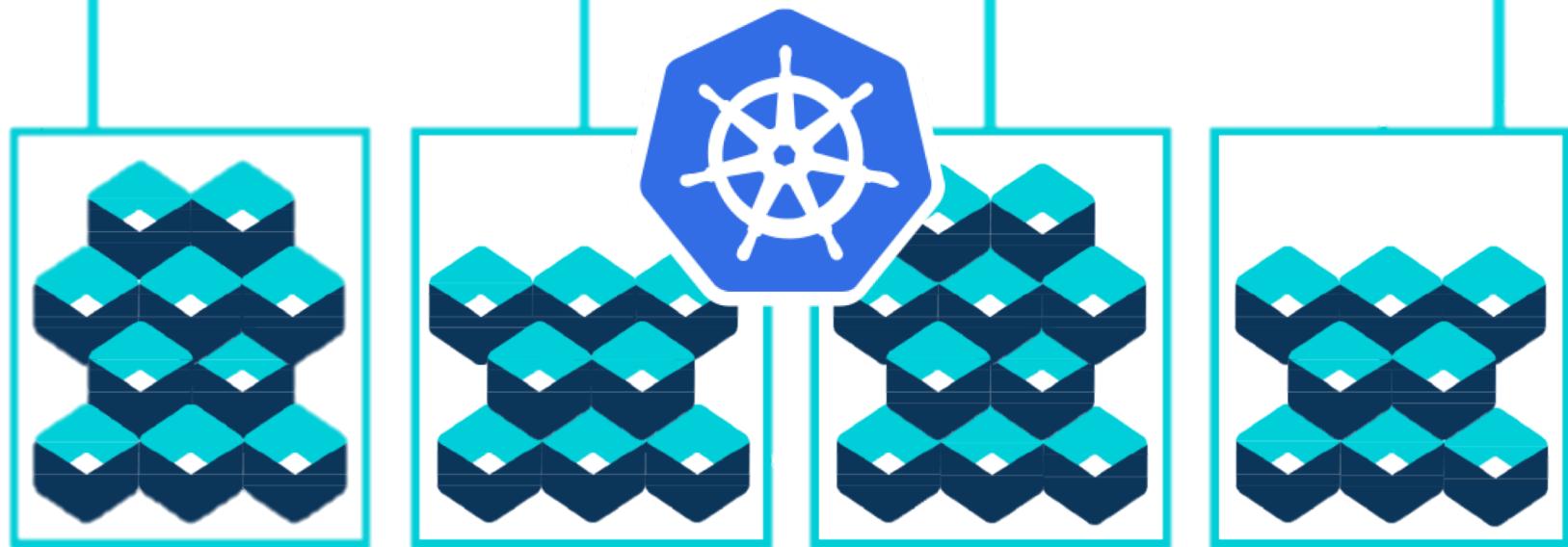




But soon you have many applications, many instances...



And that is why we have Container orchestration



What is Container Orchestration?

Container orchestration

- Manages the deployment, placement, and lifecycle of workload containers

Cluster management

- Federates multiple hosts into one target

Scheduling

- Distributes containers across node

Service discovery

- Knows where the containers are located
- Distributes client requests across the containers

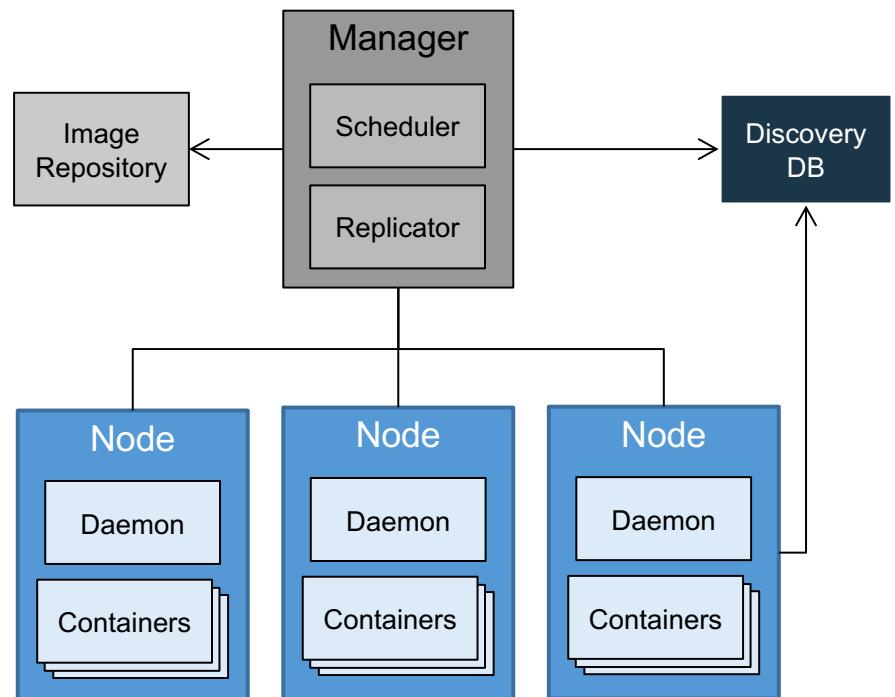
Replication

- Ensures the right number of nodes and containers

Health management

- Replaces unhealthy containers and nodes

Container Orchestrator





What is Kubernetes?

- Project started by Google
 - platform for hosting containers in a clustered environment with multiple Docker hosts
 - Provides container grouping, load balancing, auto-healing, scaling features
 - Contributors == Google, CodeOS, Redhat, Mesosphere, Microsoft, HP, IBM, VMWare, Pivotal, SaltStack, etc
-



What is Kubernetes?

- **Pod** - A group of Containers
 - **Labels** - Labels for identifying pods
 - **Kubelet** - Container Agent
 - **Proxy** - A load balancer for Pods
 - **etcd** - A metadata service
 - **cAdvisor** - Container Advisor provides resource usage/performance statistics
 - **Replication Controller** – handles pod replication
 - **Scheduler** - Schedules pods in worker nodes
 - **API Server** - Kubernetes API server
-

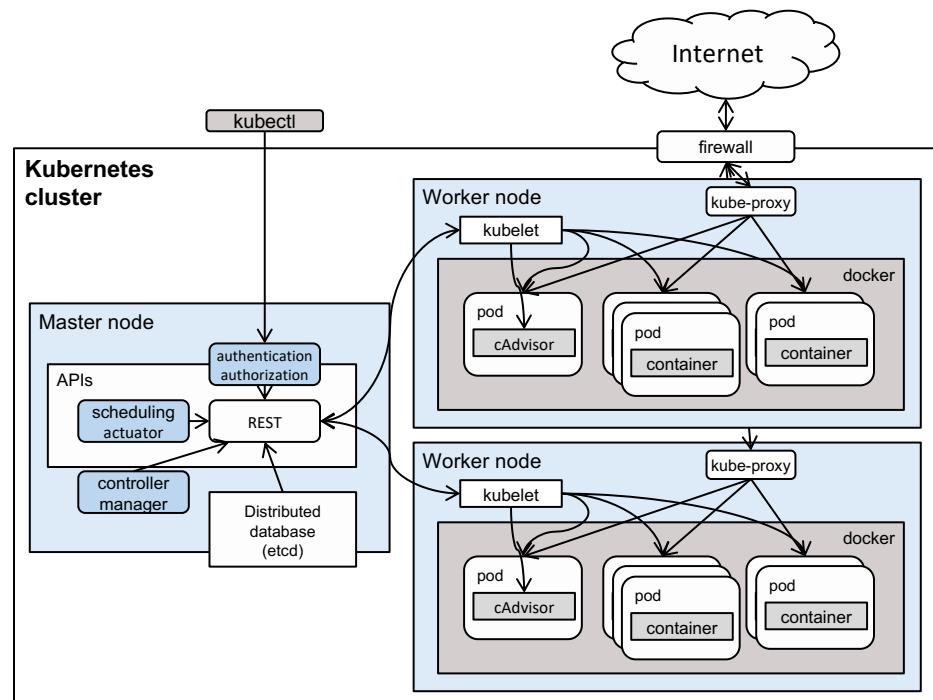
Kubernetes Cluster Architecture

Master node

- Node that manages the cluster
- Scheduling, replication & control
- Multiple nodes for HA

Worker nodes

- Node where pods are run
- Docker engine
- kubelet agent accepts & executes commands from the master to manage pods
- cAdvisor – Container Advisor provides resource usage and performance statistics
- kube-proxy – routes inbound or ingress traffic



Time to go do labs...

<https://github.com/irvnet/dallas-containerWorkshop-may2018>

Thank you