

Intro to Kubernetes Workshop



MAY
31

Hands On Container Lab
with Kubernetes

Richard Irving

Agenda

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

The Objective

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

Getting Started...

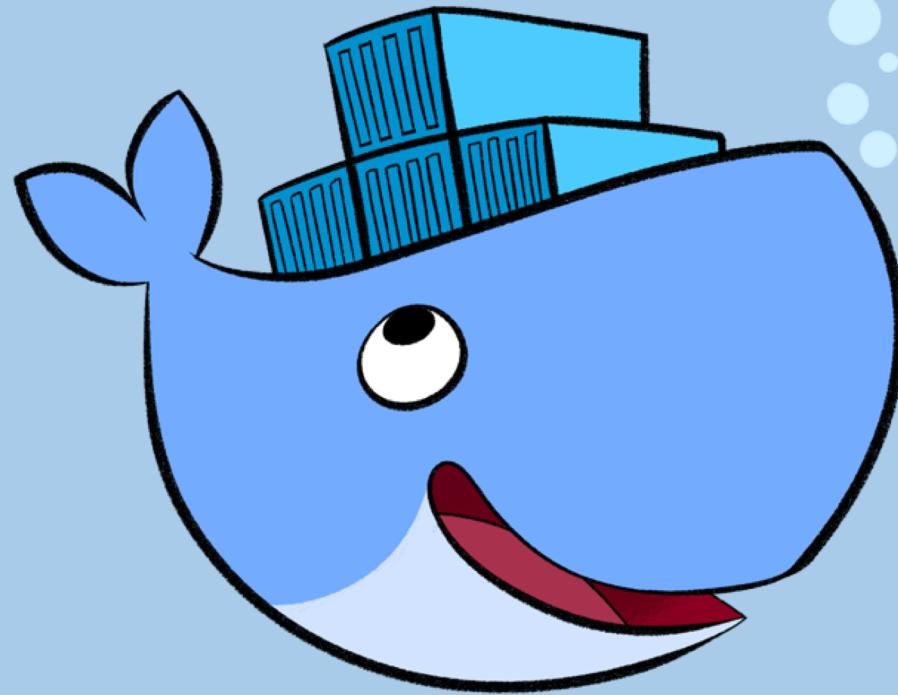
1. *Cluster URL*
2. *Username & password*
3. *Download and configure the [kubectl](#) client*
4. *Create an account at <https://hub.docker.com>*

Where are the labs?



https://github.com/kriersd/ICP_Meetup_Workshop

What is Docker?



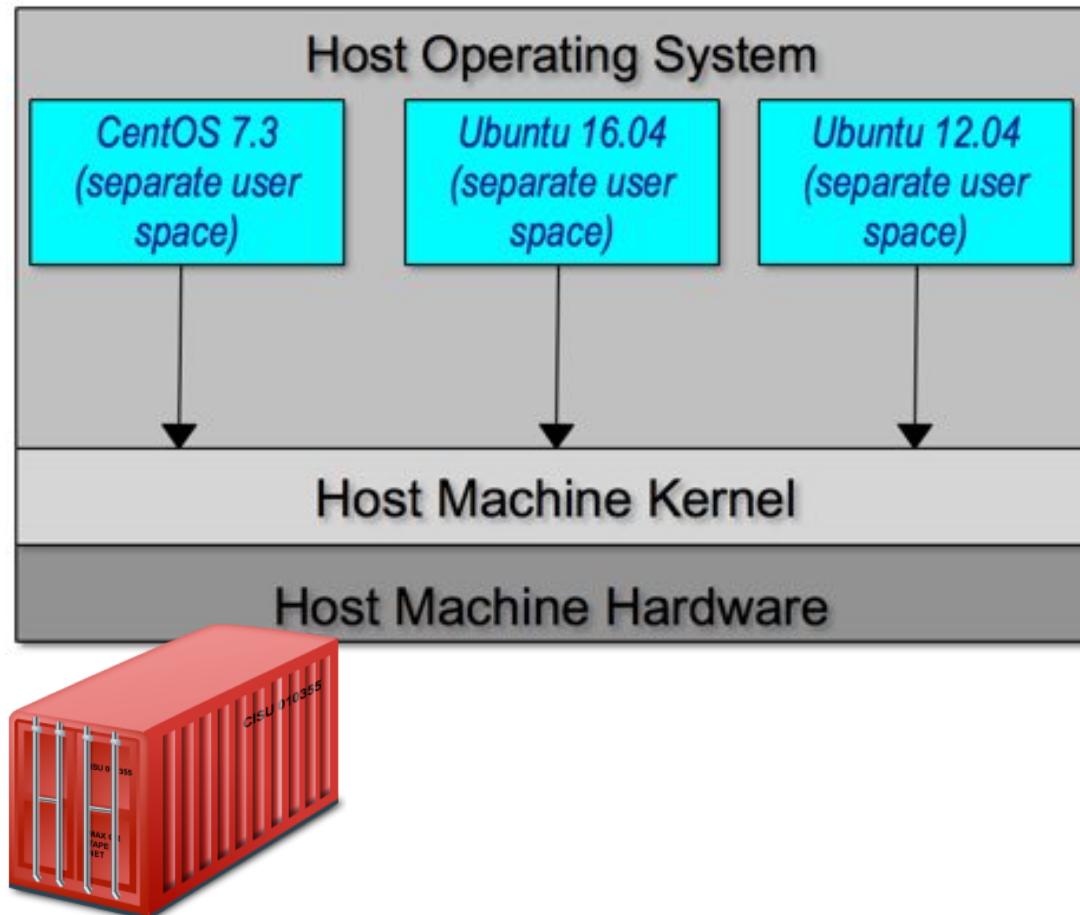
Containers VS Virtual machines



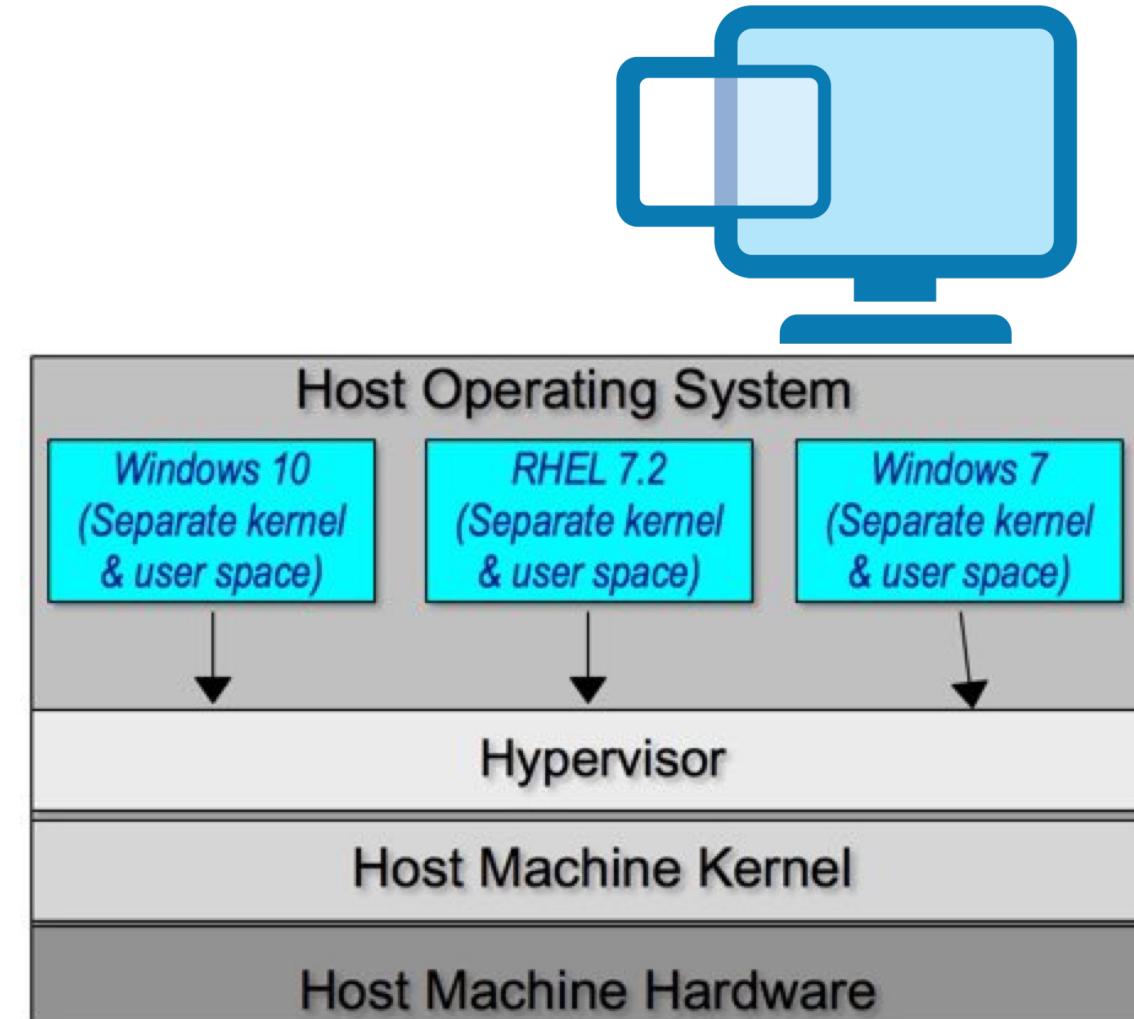
VS



Containers VS Virtual machines

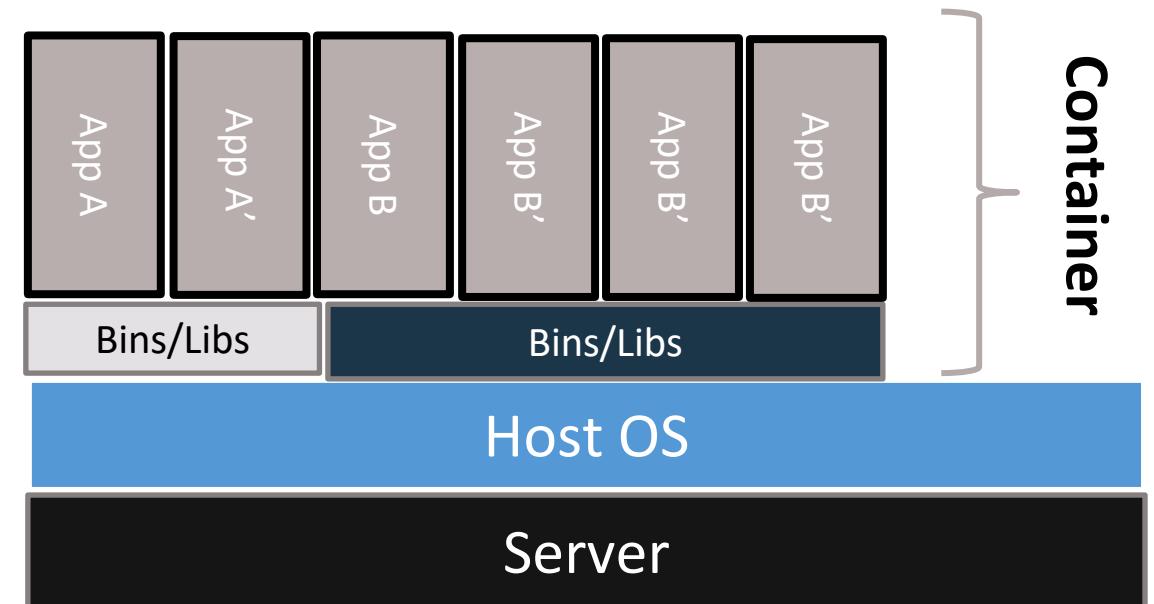


VS



What's a container anyway?

- 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



Separation of Concerns

Debbie: The Developer

Handles what's "inside" the container

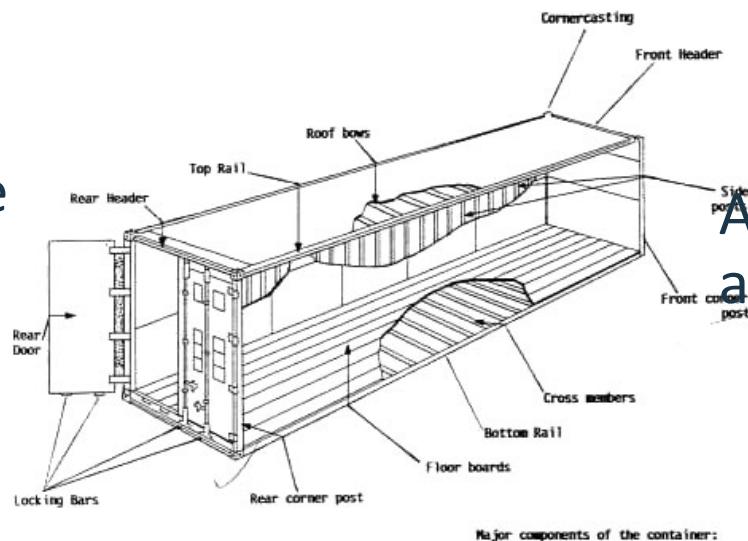
- Her code
- Her Libraries
- Her Package Manager
- Her Apps
- Her Data

Linux servers all look the same

Oswaldo: The Ops Guy

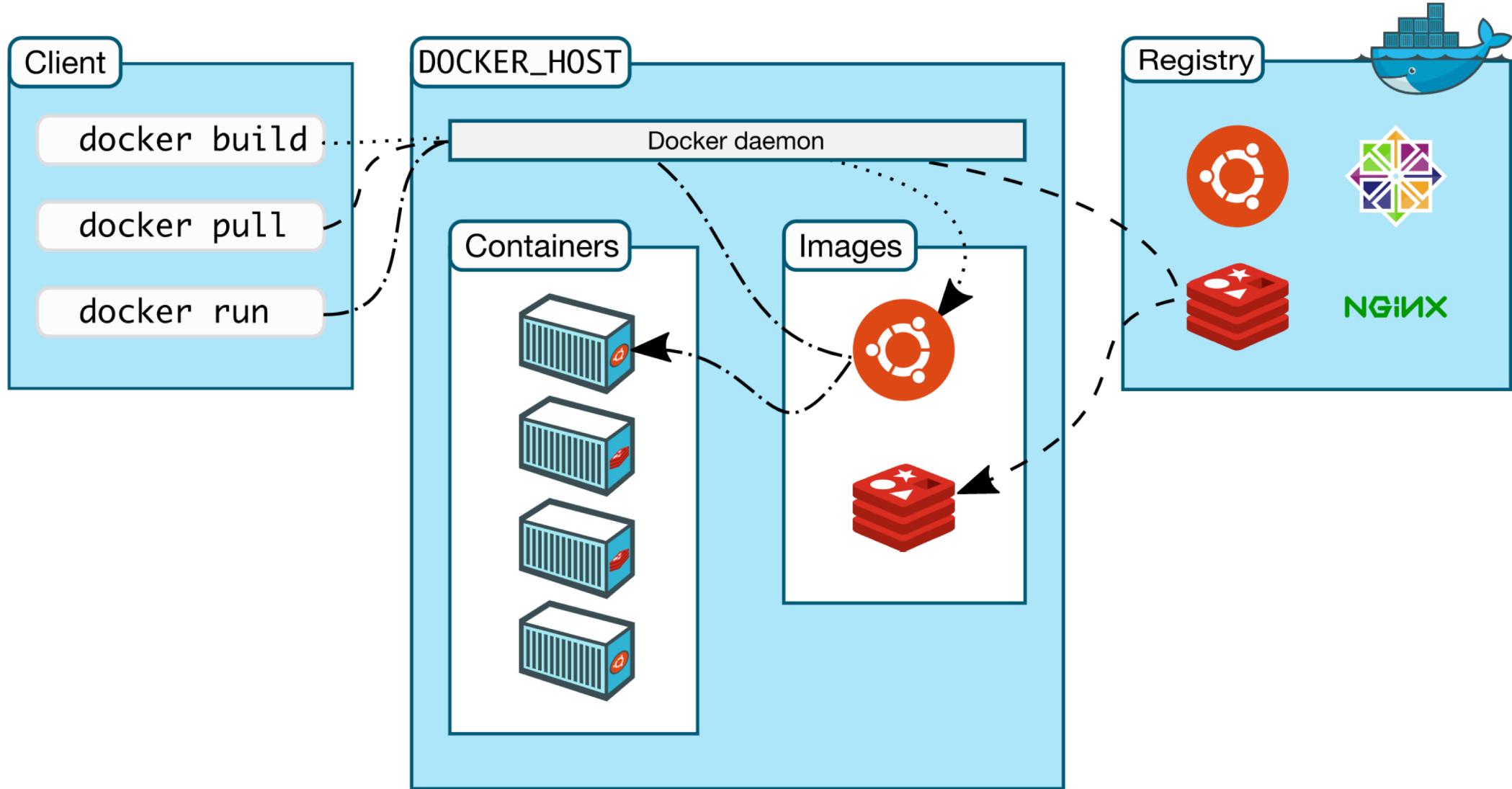
Handles what's "outside" the container

- Logging
- Remote access
- Monitoring
- Network configuration



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

Docker Architecture



Images vs Containers



Image

An inert, immutable, file that's essentially a snapshot of a container.

Are created with the build command, and they'll produce a container when started with run.

Images are stored in a Docker registry such as registry.hub.docker.com.

Are designed to be composed of layers of other images, allowing a minimal amount of data to be sent when transferring images over the network.

To list images

```
# docker images
```



Containers are mortal, live and die

Not restarted

New container instance started using image

If persistence is required external storage is employed

To list containers

```
# docker ps (Running)
```

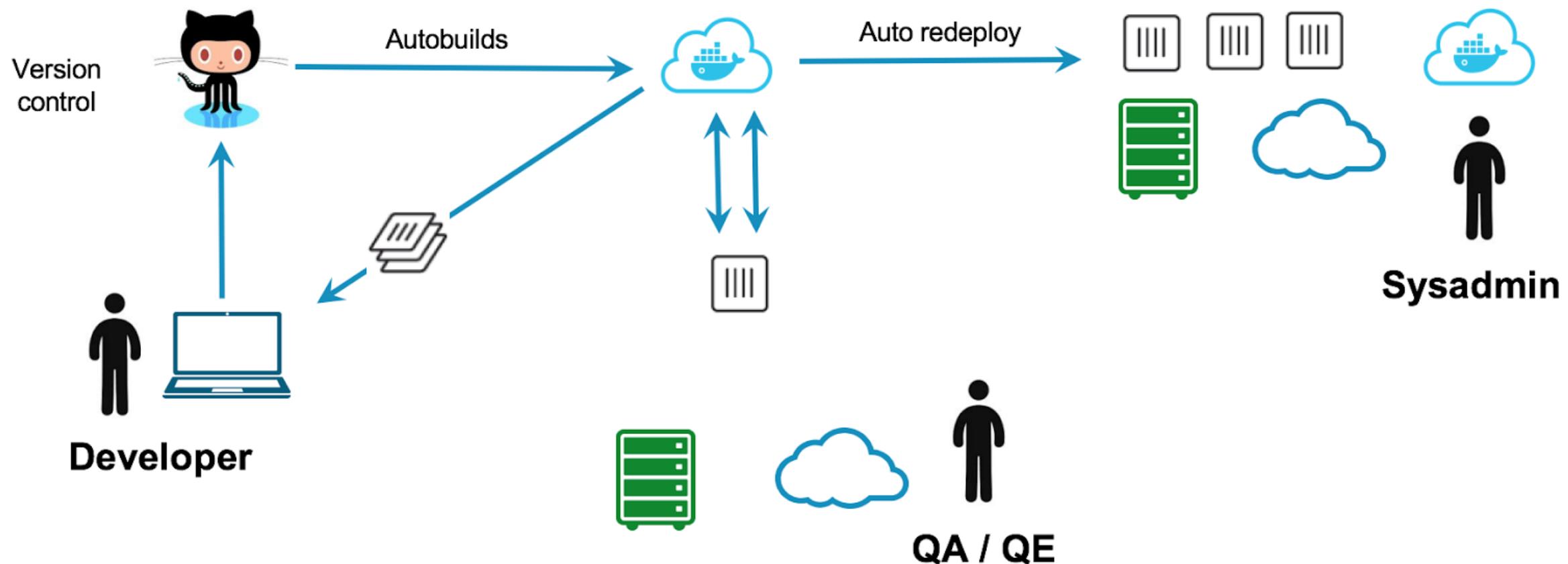
```
# docker ps -a (All)
```

Docker Workflow

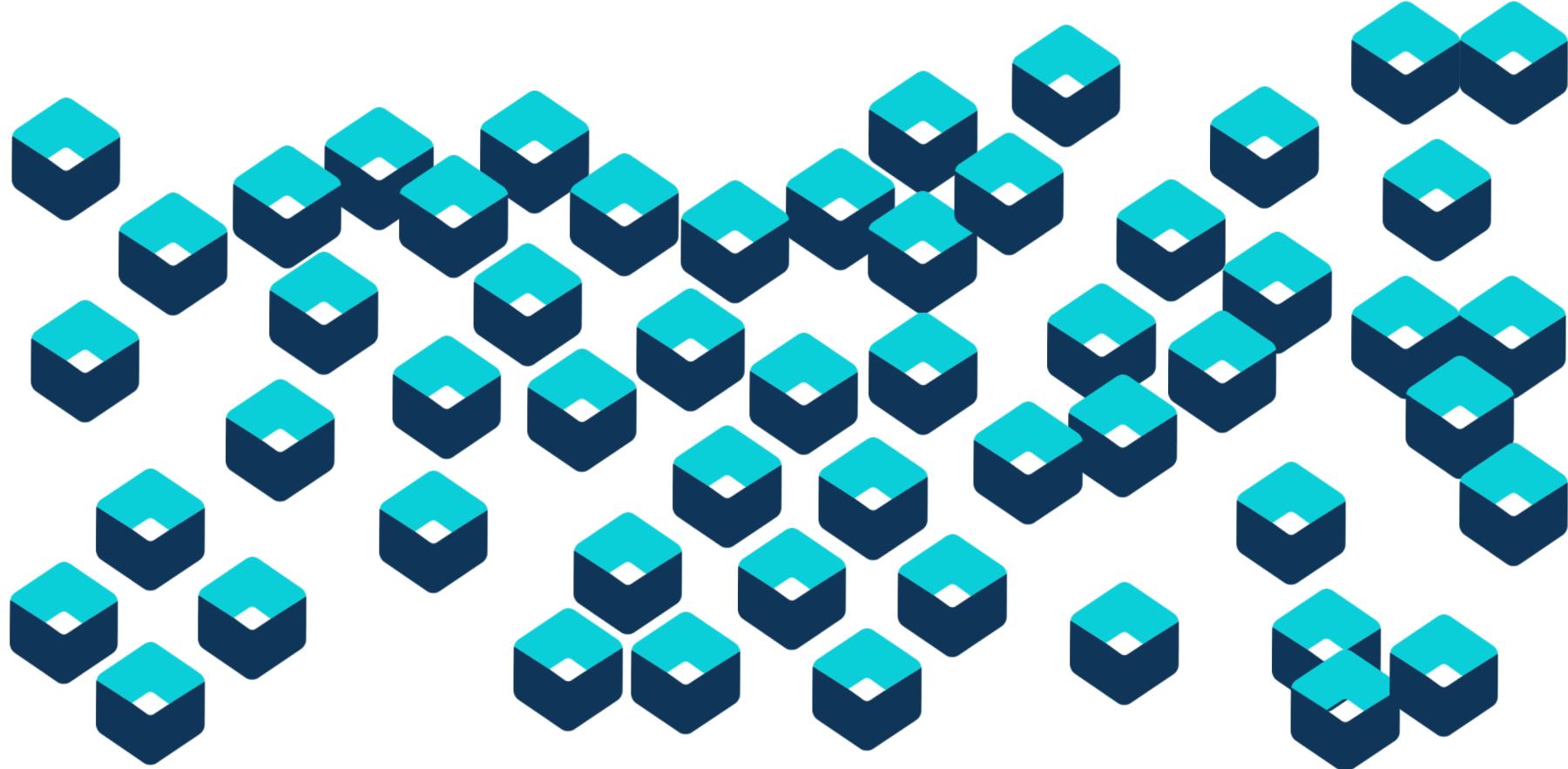
1. Development

2. Test

3. Stage / Production



Containers are great but ... can lead into lack of control & chaos



What is Kubernetes?





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 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

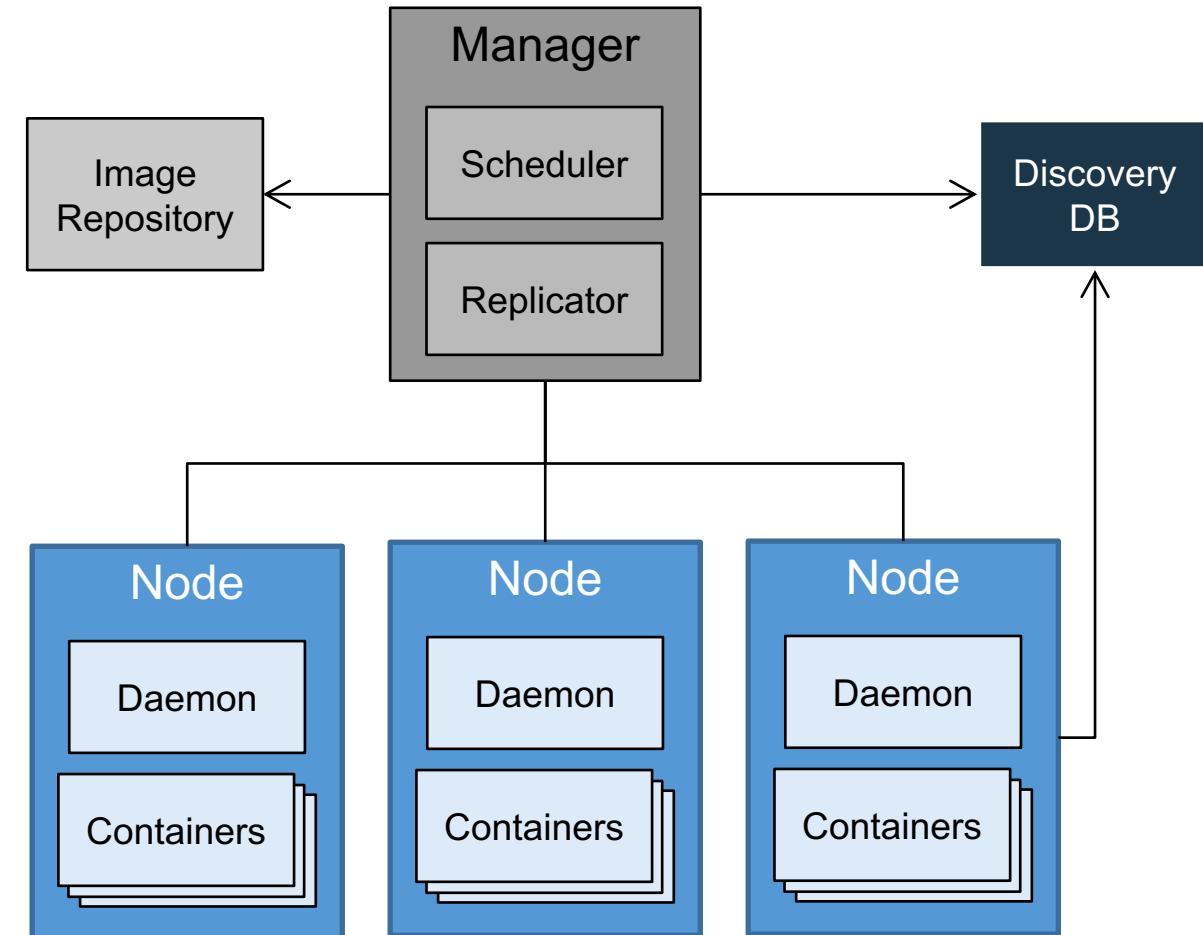
Replication

- Ensures the right number of nodes and containers

Health management

- Replaces unhealthy containers and nodes

Container Orchestrator



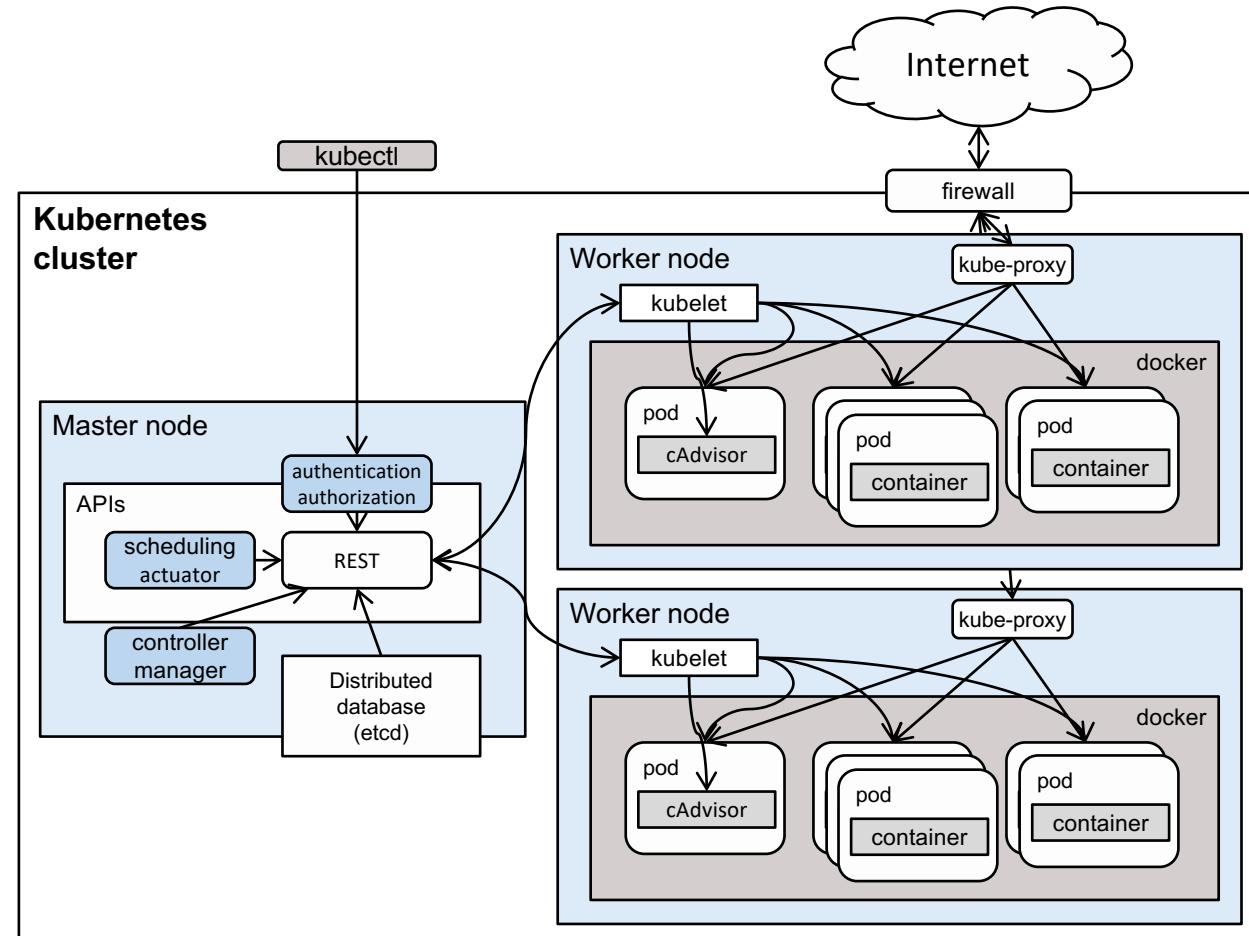
Kubernetes Cluster Architecture

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



Kubernetes Objects...

- **Pod** - represents a unit of deployment: a single instance of an application in Kubernetes
- **Service** - abstraction which defines a logical set of Pods and a policy by which to access
- **Volume** - makes data persistent though pods are not...
- **Namespace** - A scope for names, and a mechanism to attach authorization and policy to a subsection of the cluster.

- **Deployment** - describe a desired state in a Deployment object, and the Deployment controller changes the actual state at a controlled rate
- **ReplicaSet** - ensures that a specified number of pod replicas are running at any given time.
- **StatefulSet** - maintains a sticky identity for each of their Pods. Pods are from the same spec, but each has a persistent identifier that maintains across any rescheduling.
- **Job** - A job is a supervisor for pods carrying out batch processes

Hey... psst!

Wanna see some docker stuff?

Time to go do labs...

https://github.com/kriersd/ICP_Meetup_Workshop



Additional Resources

IBM Cloud Private Hosted Trial

A fully hosted Kubernetes private cloud environment for evaluation and learning.

<https://www.ibm.com/cloud/garage/tutorials/ibm-cloud-private-trial/ibm-cloud-private-hosted-trial#accessing-the-environment>

IBM Cloud Private

Fast. Flexible. Enterprise-grade.

DTE

<https://ibm-dte.mybluemix.net/ibm-cloud-private>

Welcome to developerWorks

Find tutorials, tools, and communities for developers

<https://www.ibm.com/developerworks/>

CALL FOR CODE™

https://developer.ibm.com/callforcode/?cm_sp=dw-callforcode_-_homepage_-_overlay

Thank you