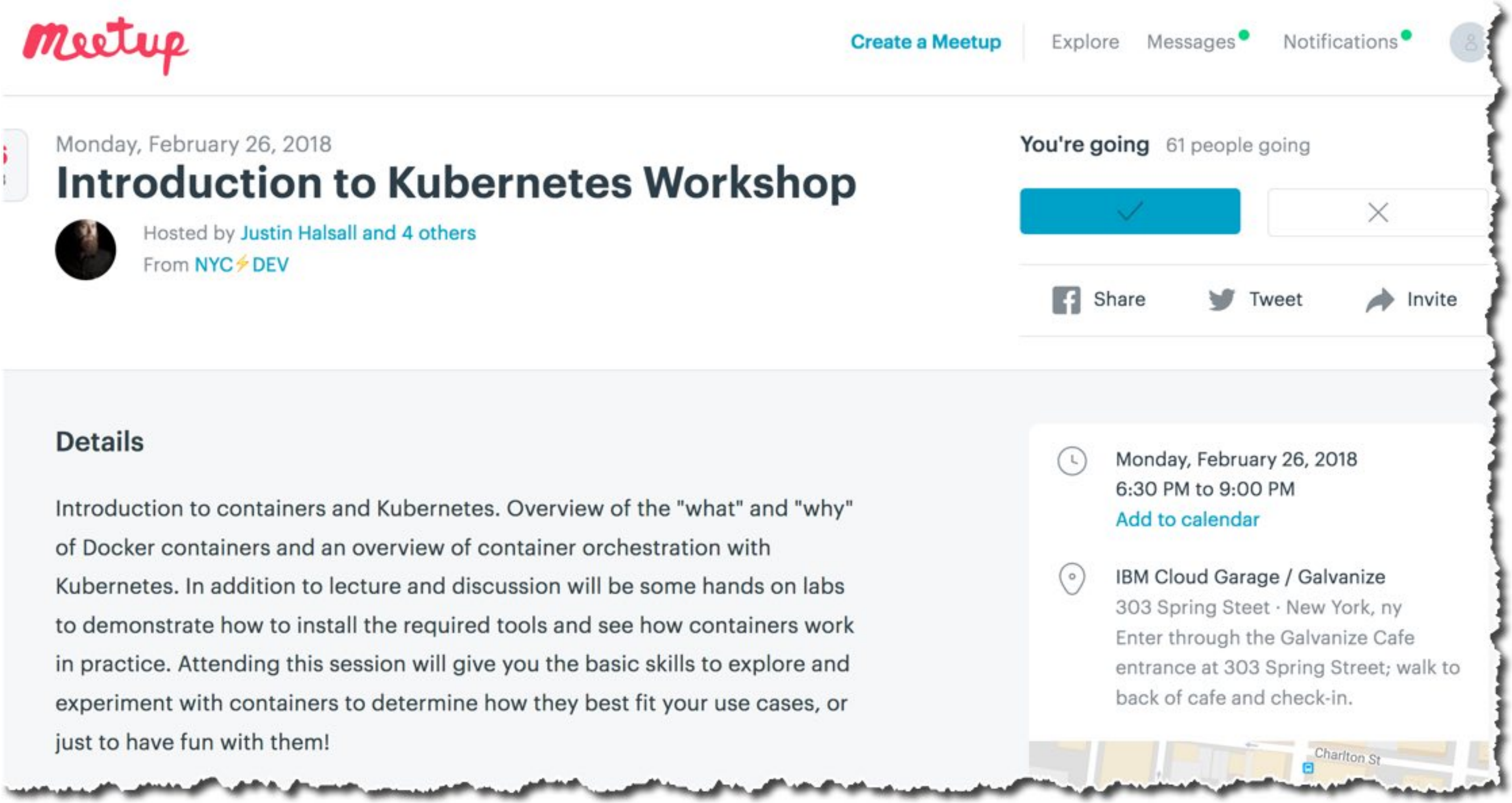
Intro to Kubernetes Workshop!



Monday, February 26, 2018

Introduction to Kubernetes Workshop

Hosted by Justin Halsall and 4 others

From NYC⚡DEV

You're going 61 people going

Details

Introduction to containers and Kubernetes. Overview of the "what" and "why" of Docker containers and an overview of container orchestration with Kubernetes. In addition to lecture and discussion will be some hands on labs to demonstrate how to install the required tools and see how containers work in practice. Attending this session will give you the basic skills to explore and experiment with containers to determine how they best fit your use cases, or just to have fun with them!

Monday, February 26, 2018
6:30 PM to 9:00 PM

IBM Cloud Garage / Galvanize
303 Spring Steet · New York, ny
Enter through the Galvanize Cafe entrance at 303 Spring Street; walk to back of cafe and check-in.

Agenda

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

The Objective:

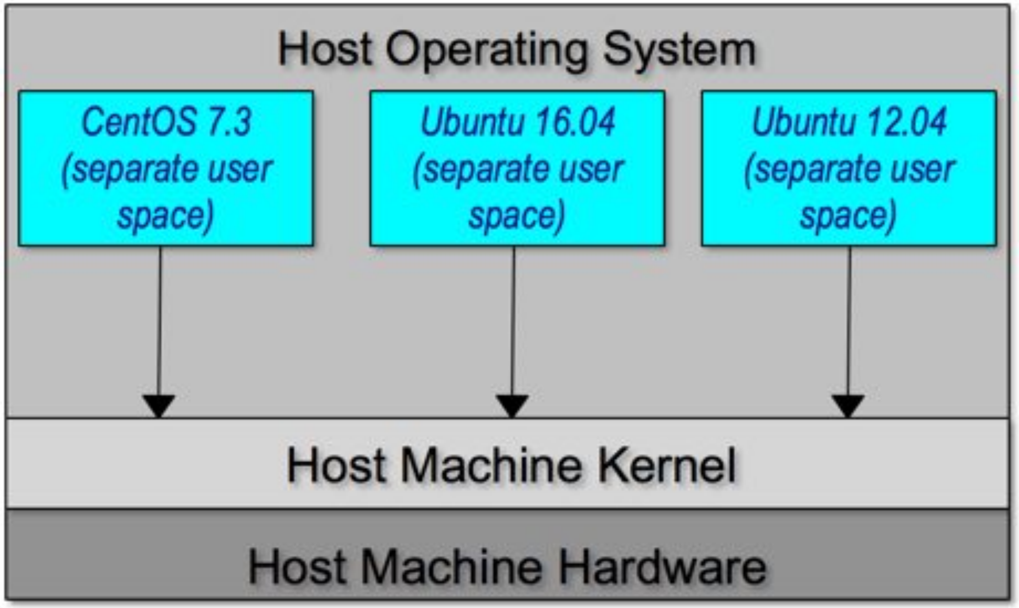
Get enough information and insight to begin experimenting with your own containerized workloads…

Where are the labs?

https://github.com/irvnet/k8s-101

What is Docker?

Containers vs VM’s…



Host Operating System

CentOS 7.3 (separate user space)

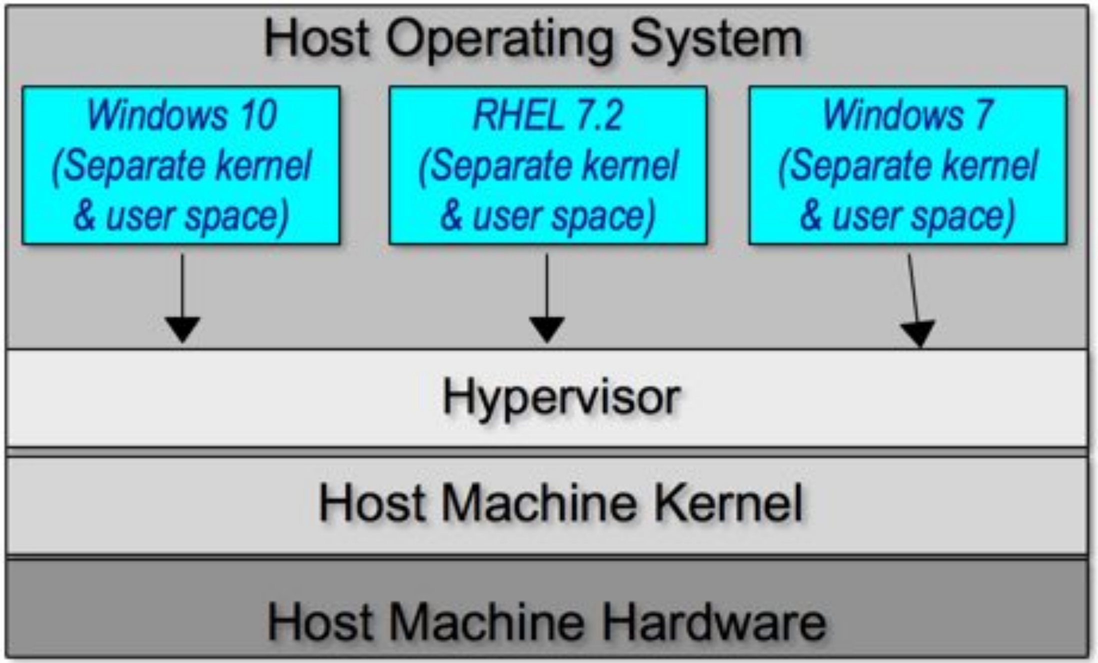
Ubuntu 16.04 (separate user space)

Ubuntu 12.04 (separate user space)

Host Machine Kernel

Host Machine Hardware

Containers vs VM's…



Host Operating System

Windows 10 (Separate kernel & user space)

RHEL 7.2 (Separate kernel & user space)

Windows 7 (Separate kernel & user space)

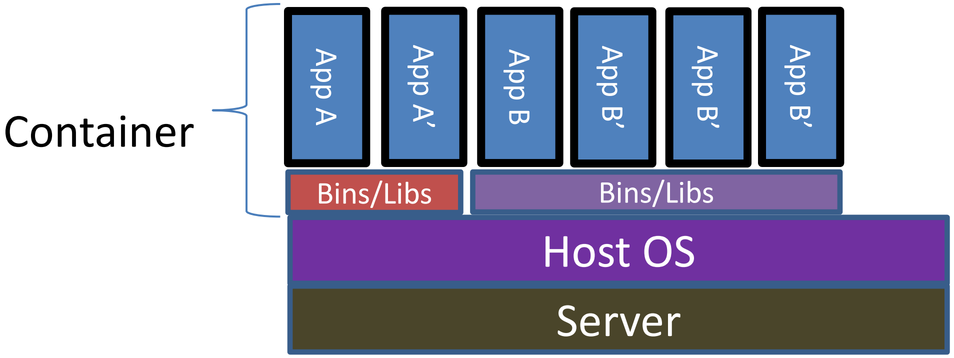
Hypervisor

Host Machine Kernel

Host Machine Hardware

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



Container

App A

App A´

App B

App B´

App B´

App B´

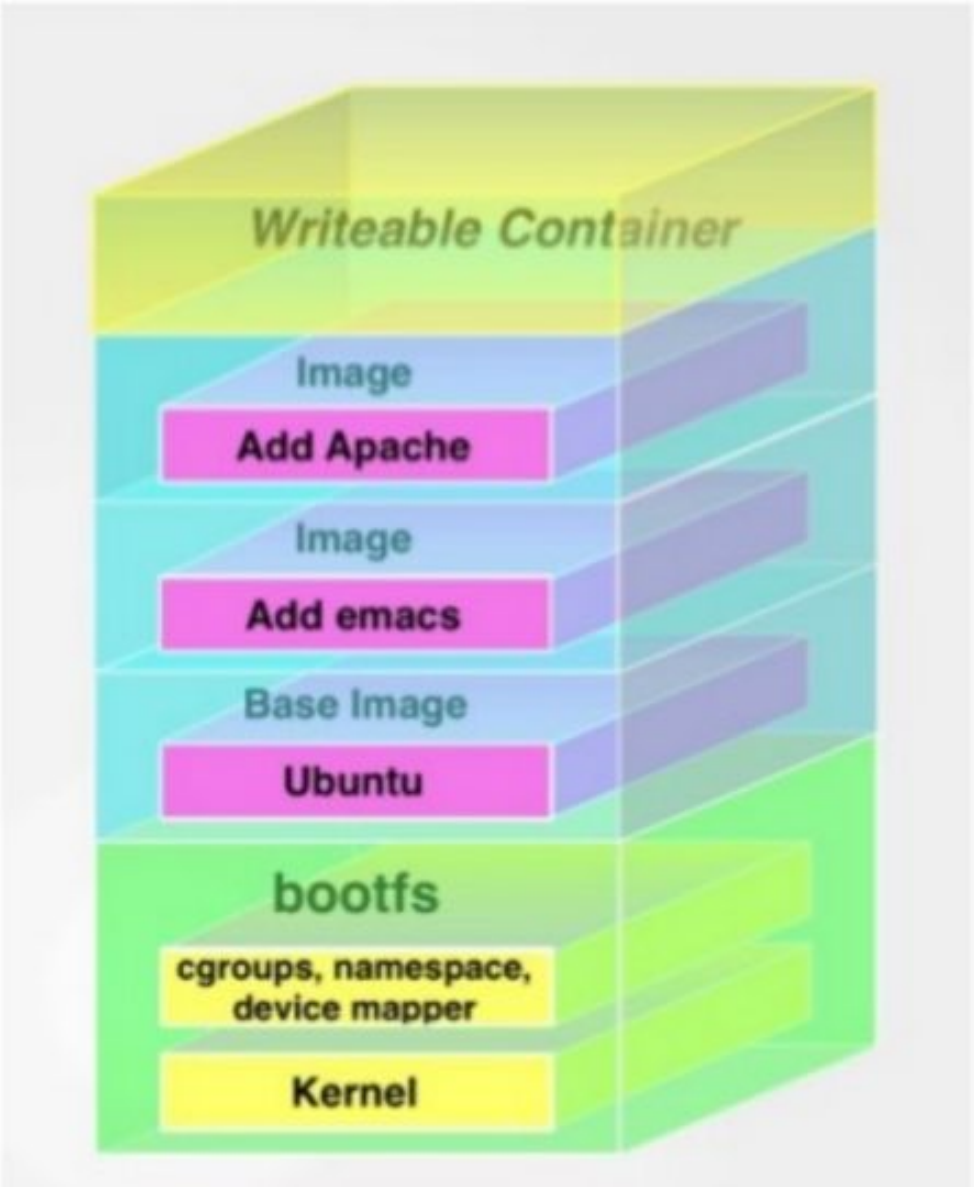
Bins/Libs

Bins/Libs

Host OS

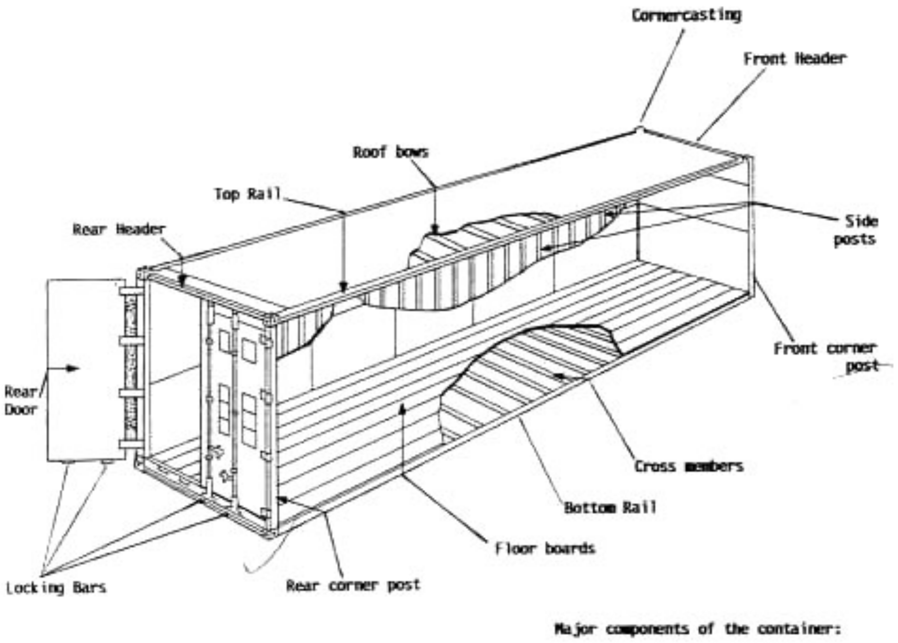
Server

Docker Image Structure



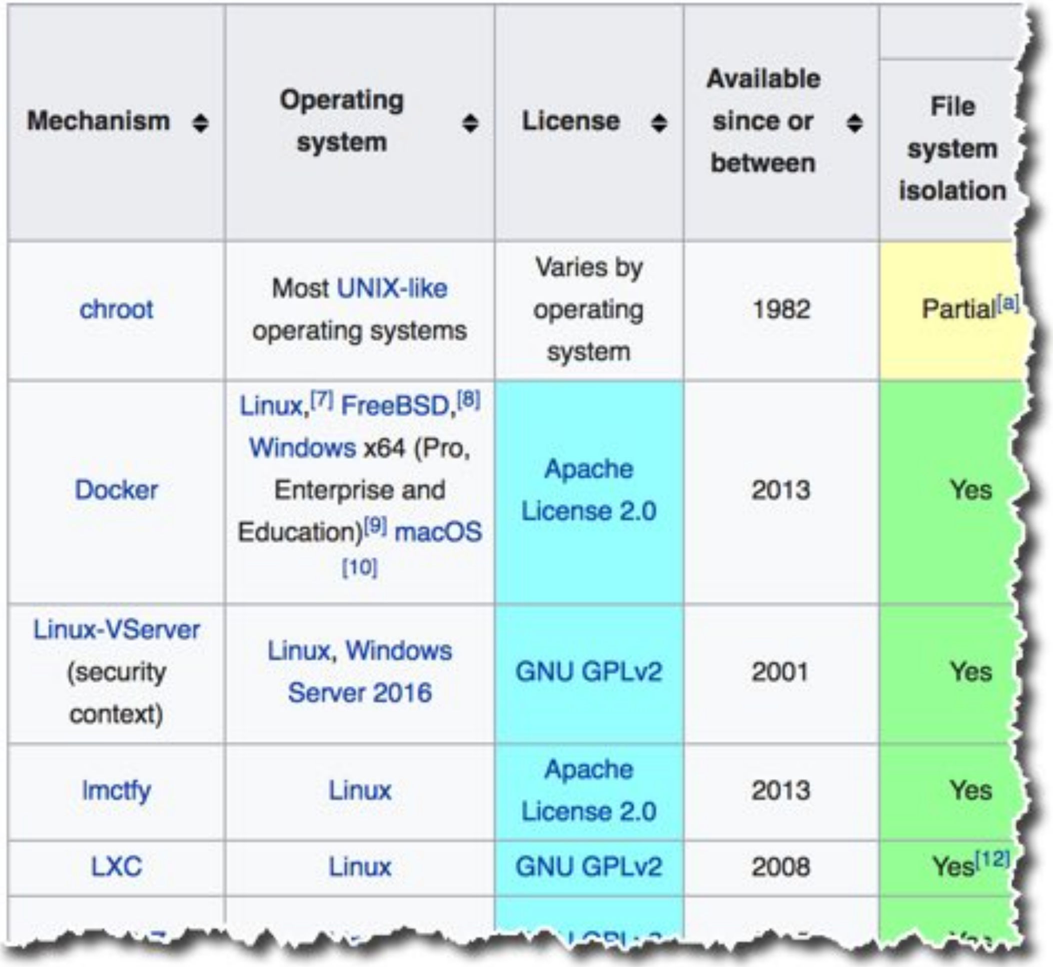
Separation of Concerns…

- **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

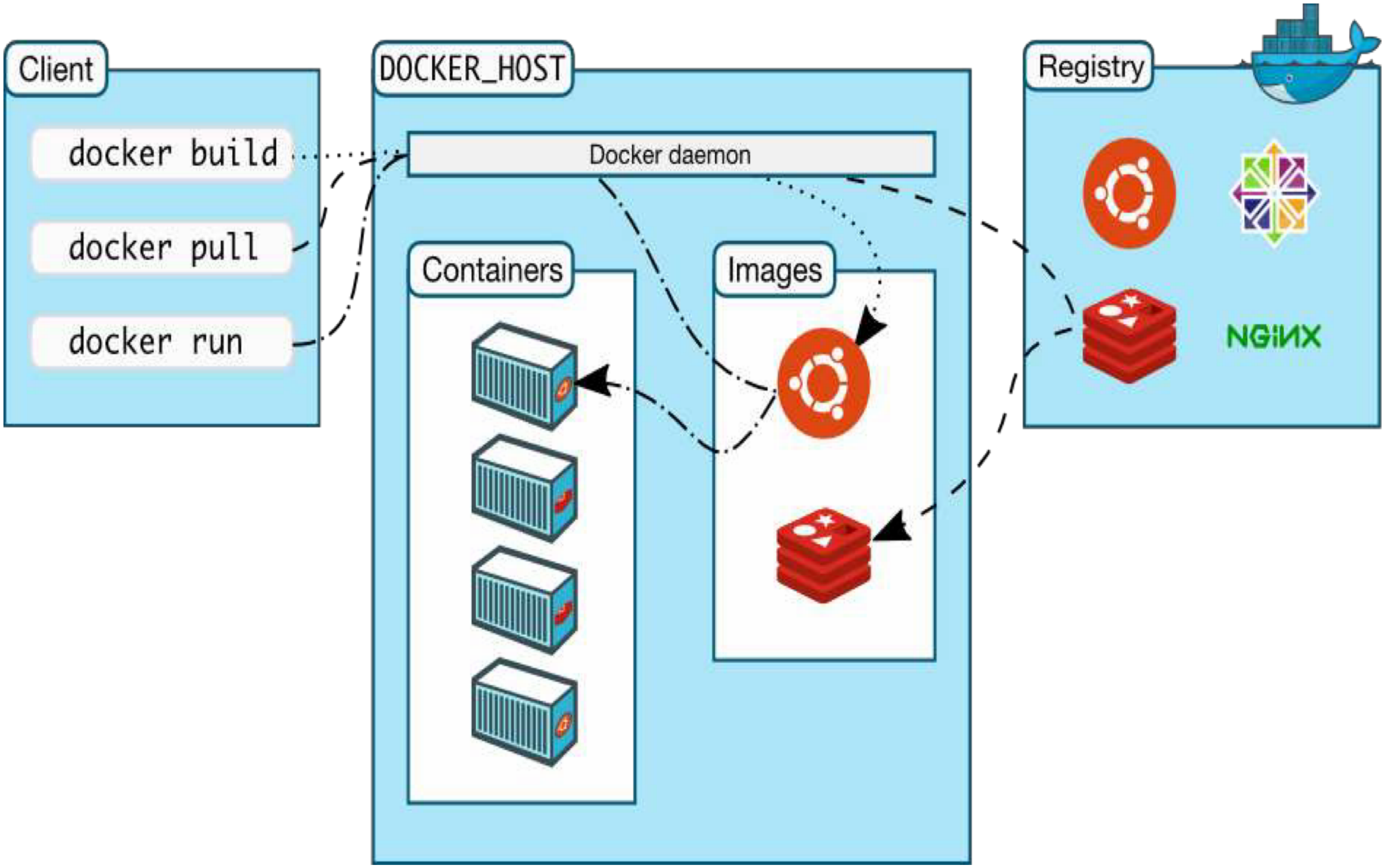
Containers weren’t born yesterday…



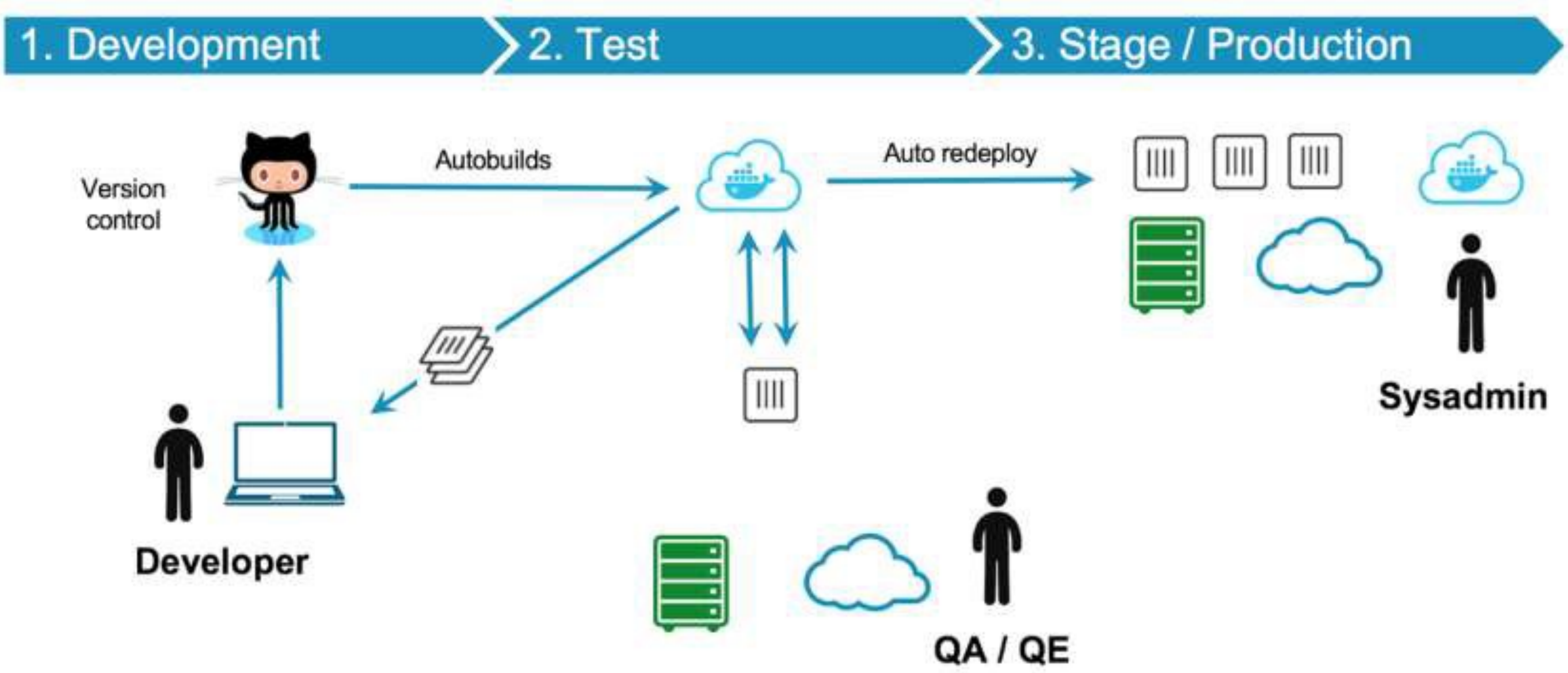
Mechanism	Operating system	License	Available since or between	File system isolation
chroot	Most UNIX-like operating systems	Varies by operating system	1982	Partial[a]
Docker	Linux,[7] FreeBSD,[8] Windows x64 (Pro, Enterprise and Education)[9] macOS [10]	Apache License 2.0	2013	Yes
Linux-VServer (security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

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

Docker Architecture



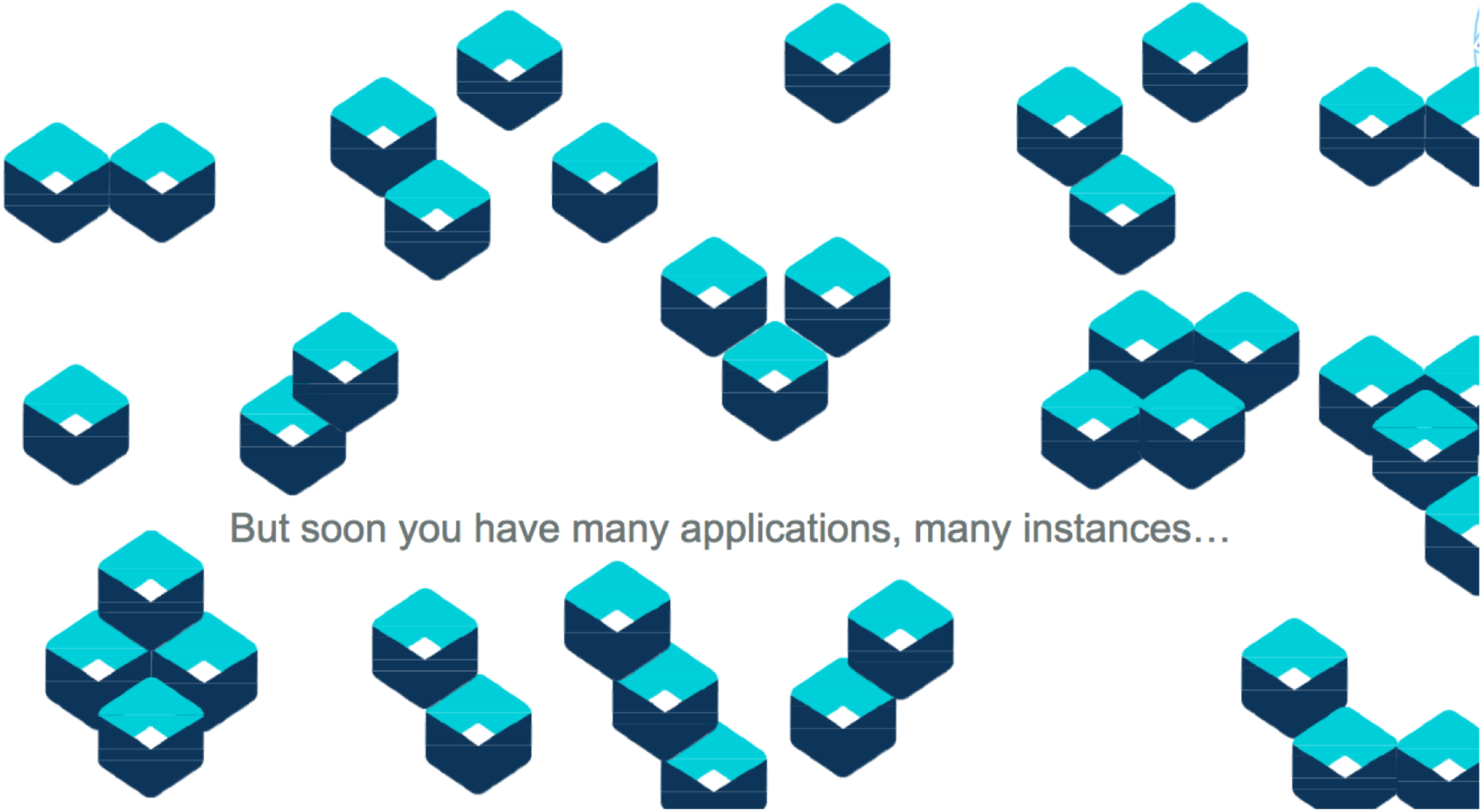
Docker Workflow



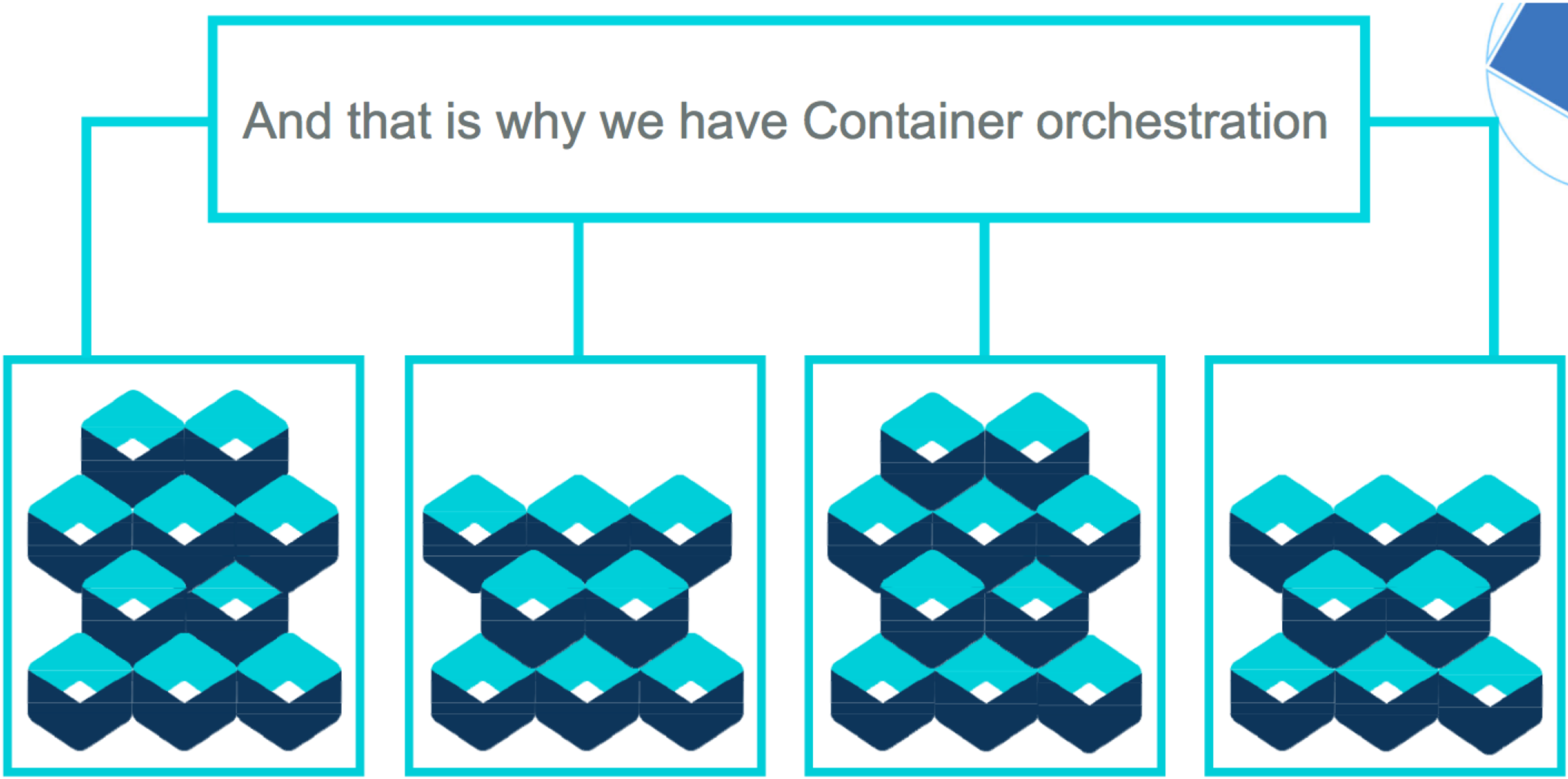
What is Kubernetes?

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 nodes

Service discovery

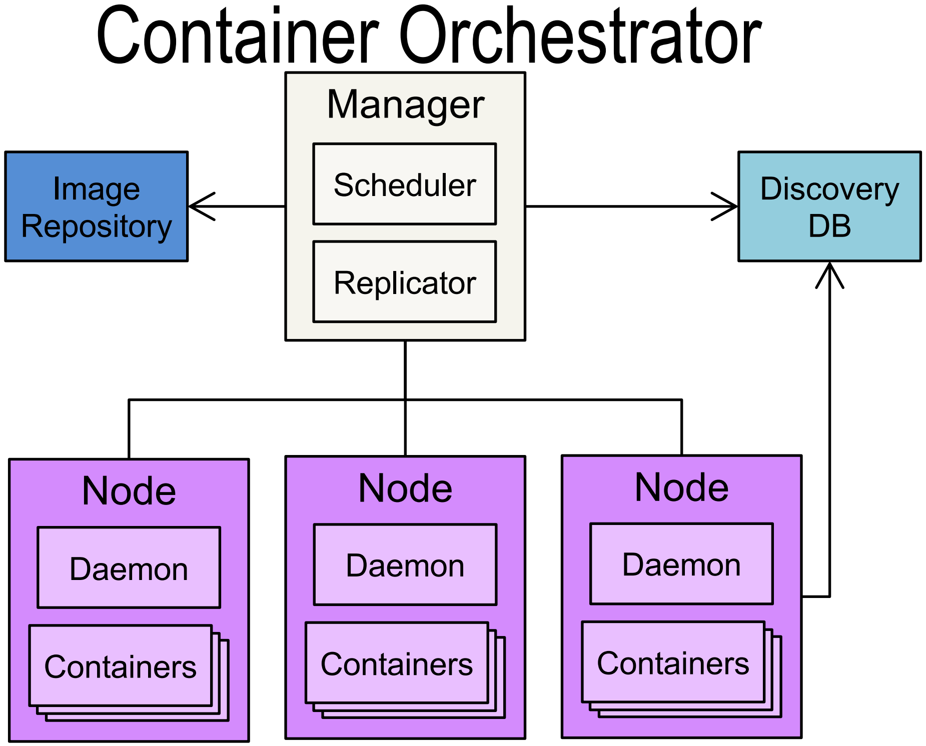
- Knows where the containers are located
- Distributes client requests across the containers

Replication

- Ensures the right number of nodes and containers

Health management

- Replaces unhealthy containers and nodes



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

Kubernetes Concepts

- **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

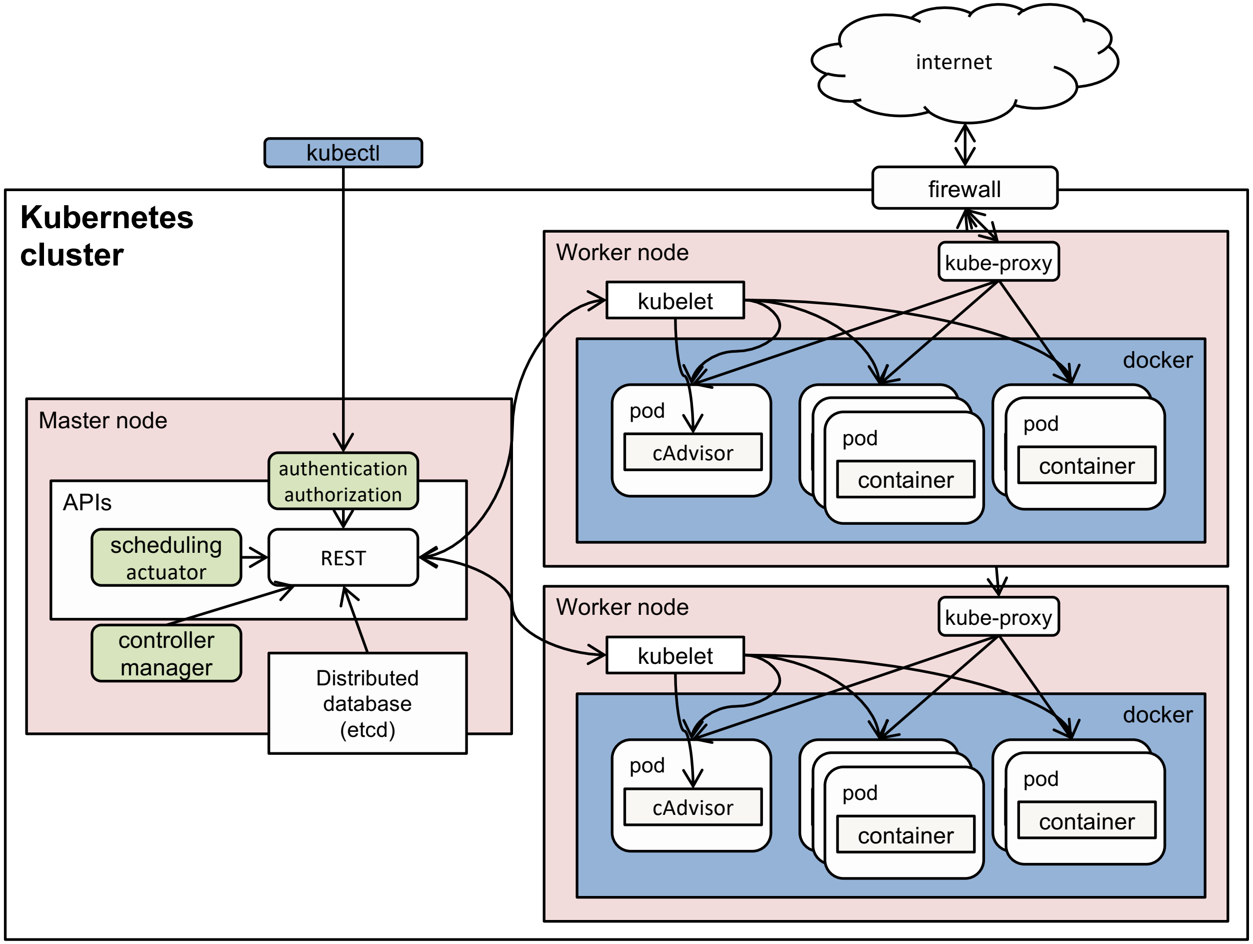
Kubernetes Cluster Architecture

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/k8s-101