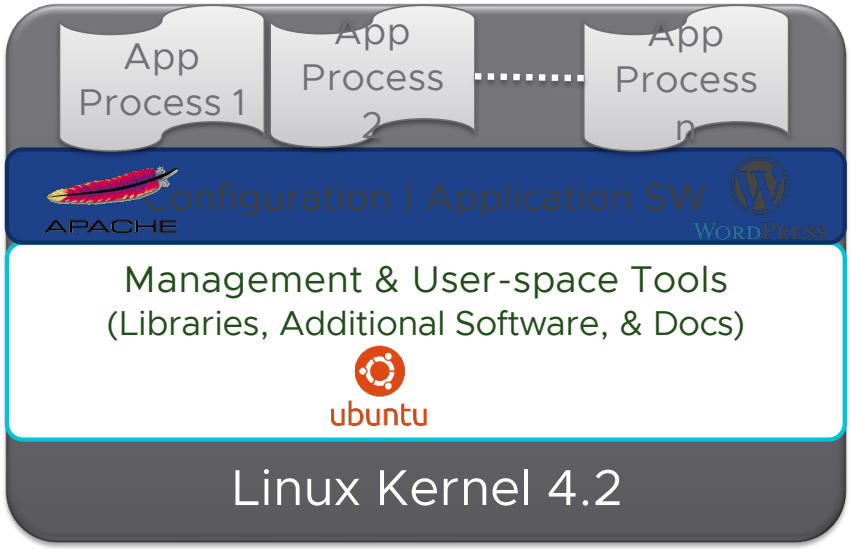
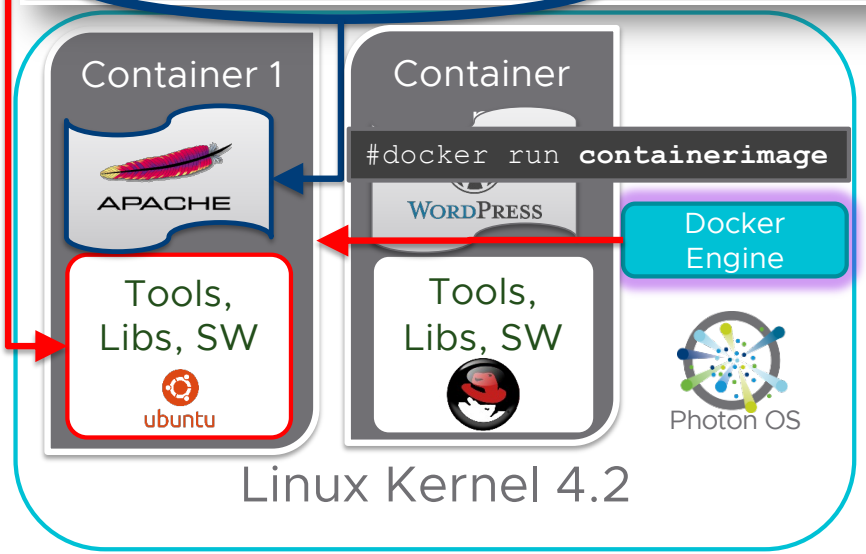
Why Use Containers?

Container image built w/Dockerfile

```
15 lines (9 sloc)  389 Bytes
1  # A basic apache server. To use either add or bind mount content under /var/www
2  FROM ubuntu:12.04
3
4  MAINTAINER Kimbro Staken version: 0.1
5
6  RUN apt-get update && apt-get install -y apache2 && apt-get clean && rm -rf /var/lib/apt/l
7
8  ENV APACHE_RUN_USER www-data
9  ENV APACHE_RUN_GROUP www-data
10 ENV APACHE_LOG_DIR /var/log/apache2
11
12 EXPOSE 80
13
14 CMD ["/usr/sbin/apache2", "-D", "FOREGROUND"]
```



Standard Linux Host



Linux "Container" Host

Challenges with Containers

CONTAINERS IN
DEVELOPMENT

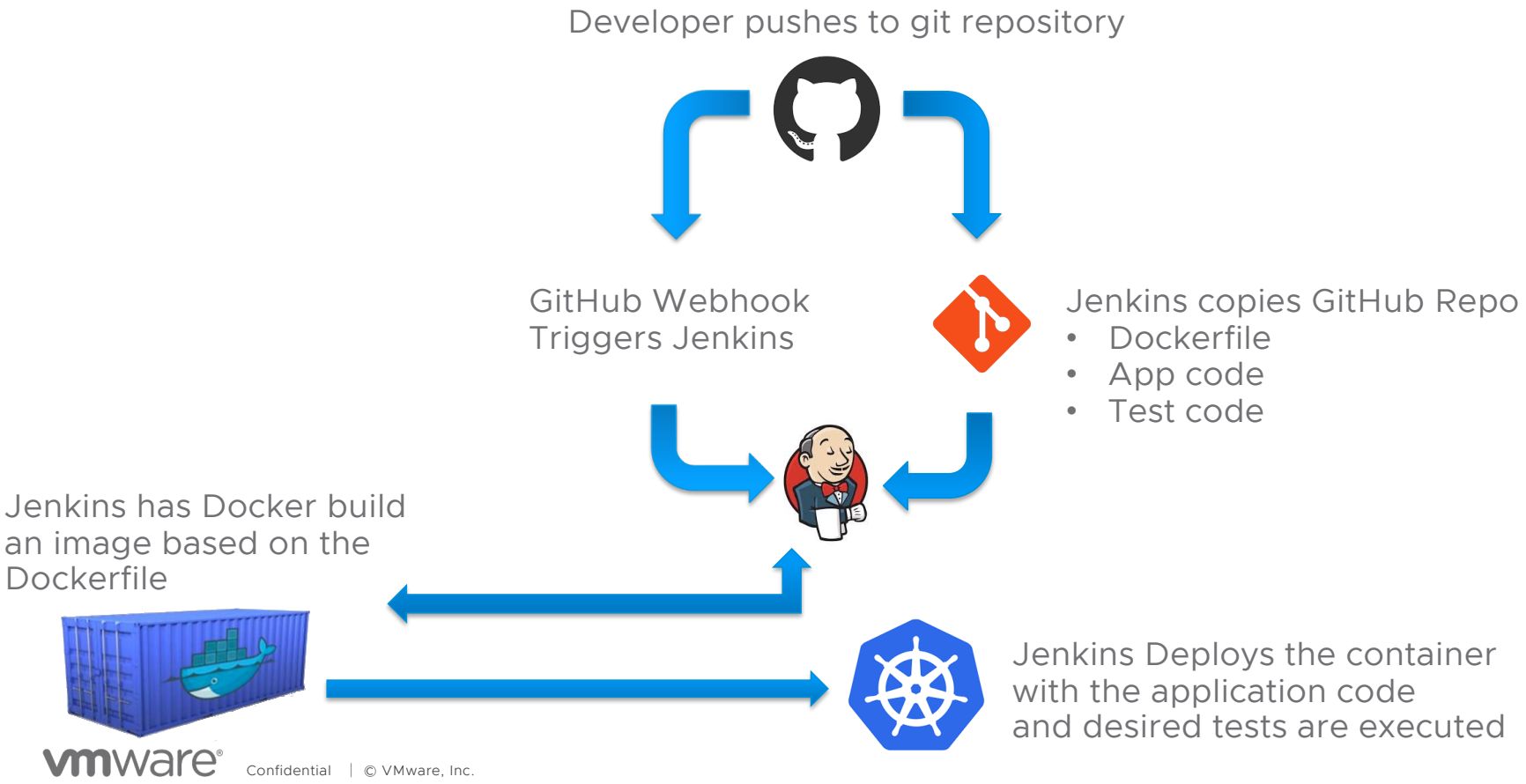
THE "LEARNING CLIFF"

Containers

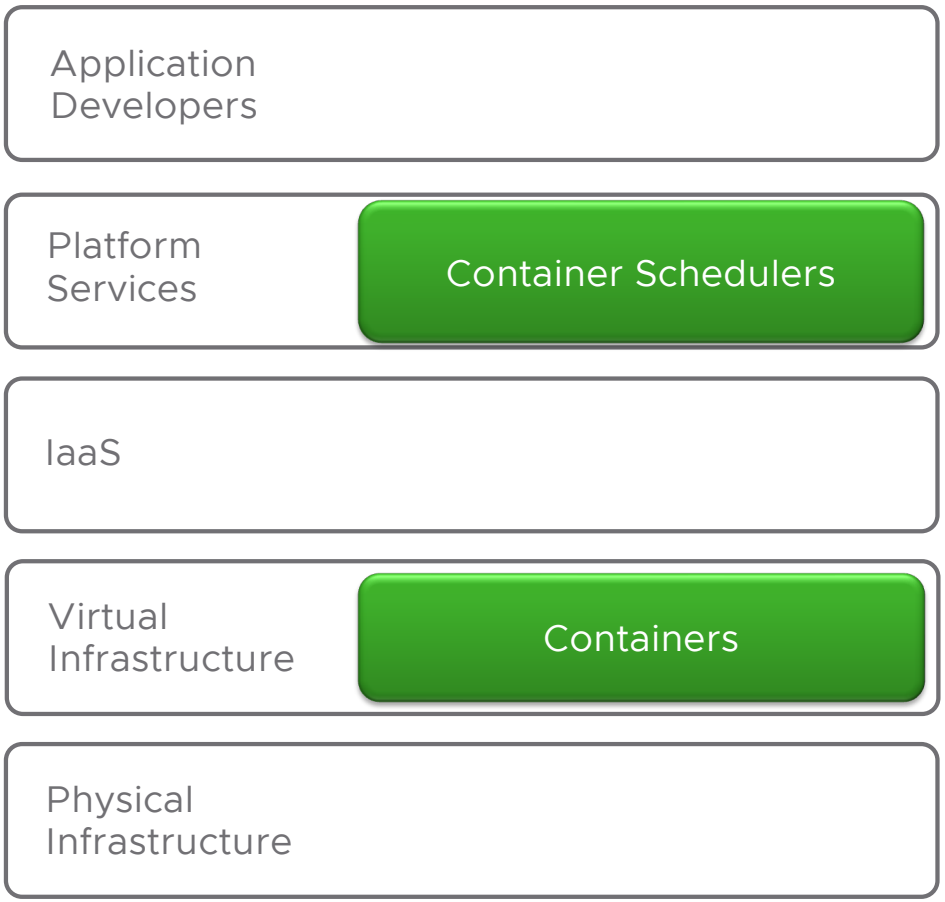
CONTAINERS IN
PRODUCTION

- Load Balancing
- Security
- High Availability
- Application Updates
- Scaling up/down
- Repeatable Deployments
- Replication
- Scheduling
- Containers

Movement from Monolithic to Microservice Architecture



Containers vs. Container Schedulers



kubernetes

Pivotal Cloud Foundry®

Introduction to Kubernetes

vmworld®

What is Kubernetes (K8s)

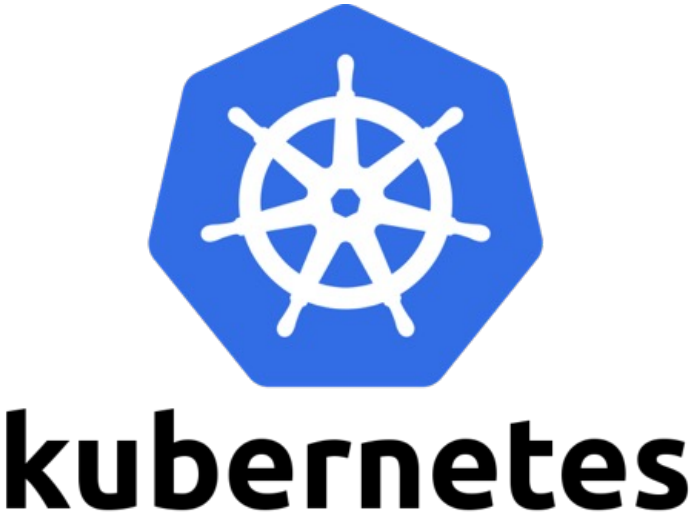
Kubernetes, is an open-source platform for managing, automating deployment, scaling, and operating containerized applications across a cluster of worker nodes.

Capabilities:

- Deploy your applications quickly and predictably
- Scale your applications on the fly
- Seamlessly roll out new features
- Optimize use of your hardware by using only the resources you need

Role:

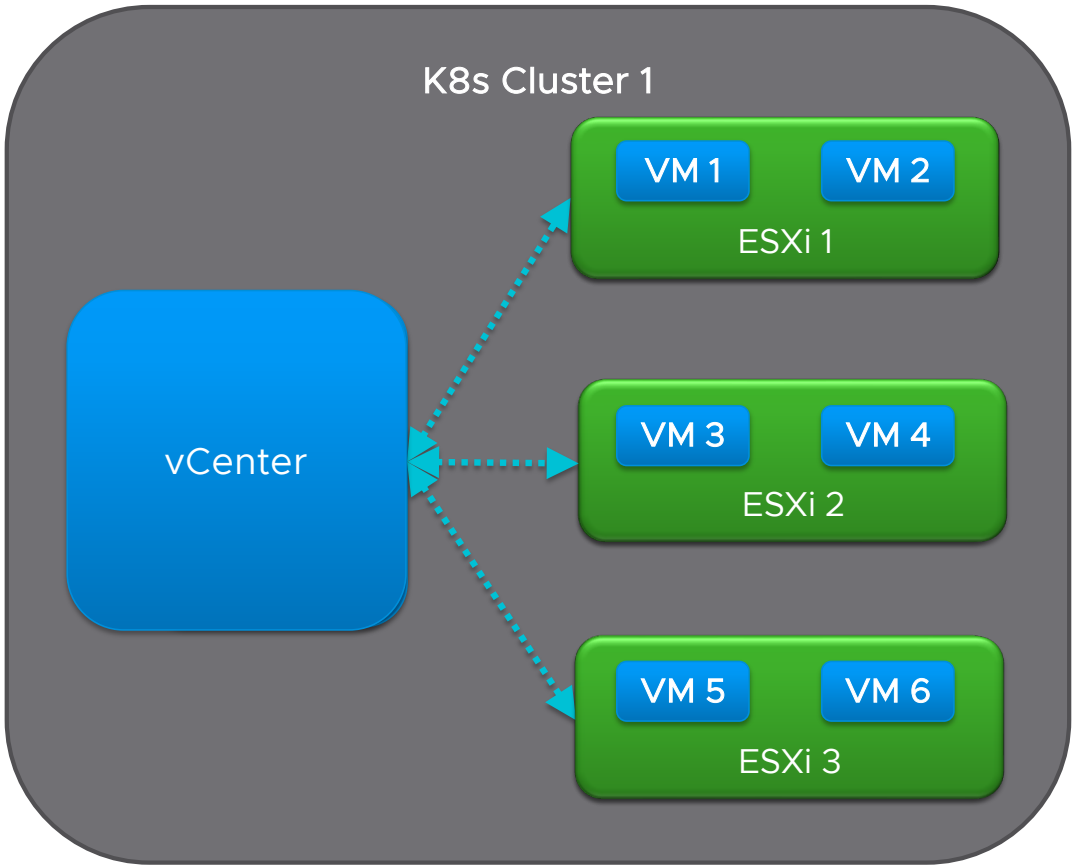
- K8s sits in the Container as a Service (CaaS) or Container orchestration layer



K8s introduces a lot new concepts

10,000 ft. View

- Cluster
- Master
- ESXi (Hosts)
- VMs



Kubernetes Core Components

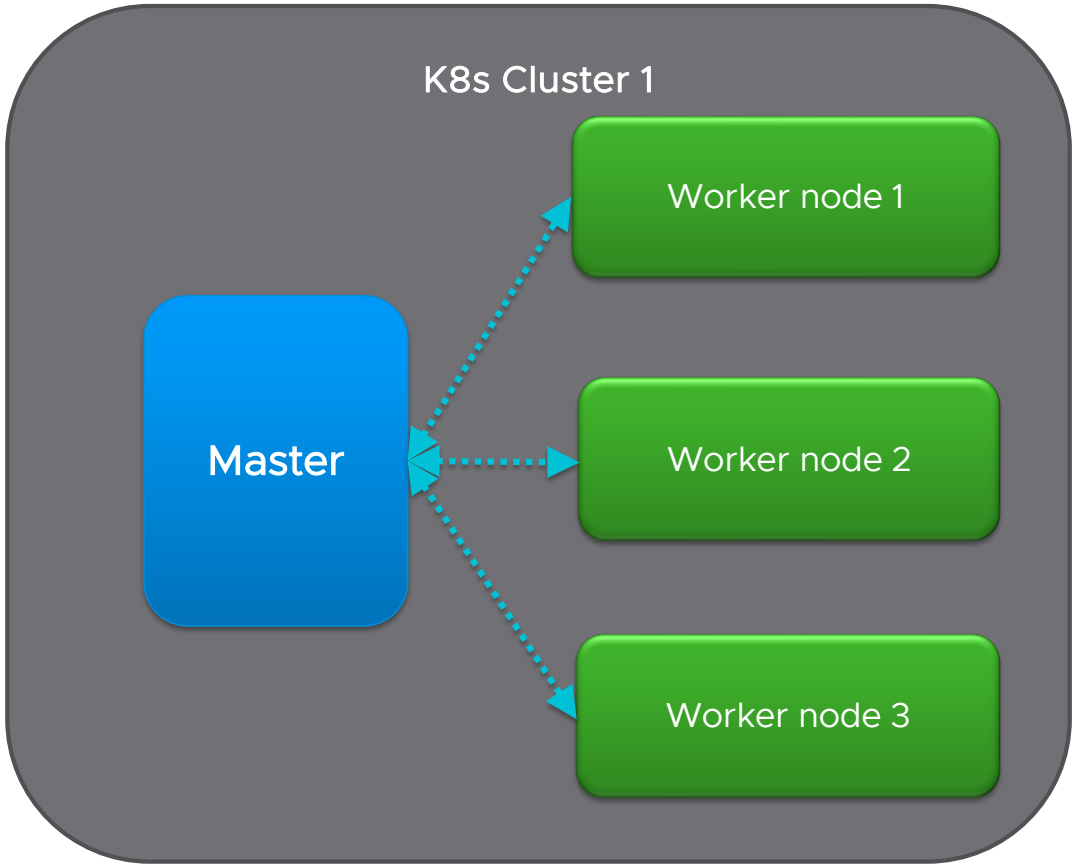
vmworld®

K8s Cluster

Consists of a master and a group of worker nodes

Namespaces

- Mechanism to partition resources into logically named groups



K8s Master

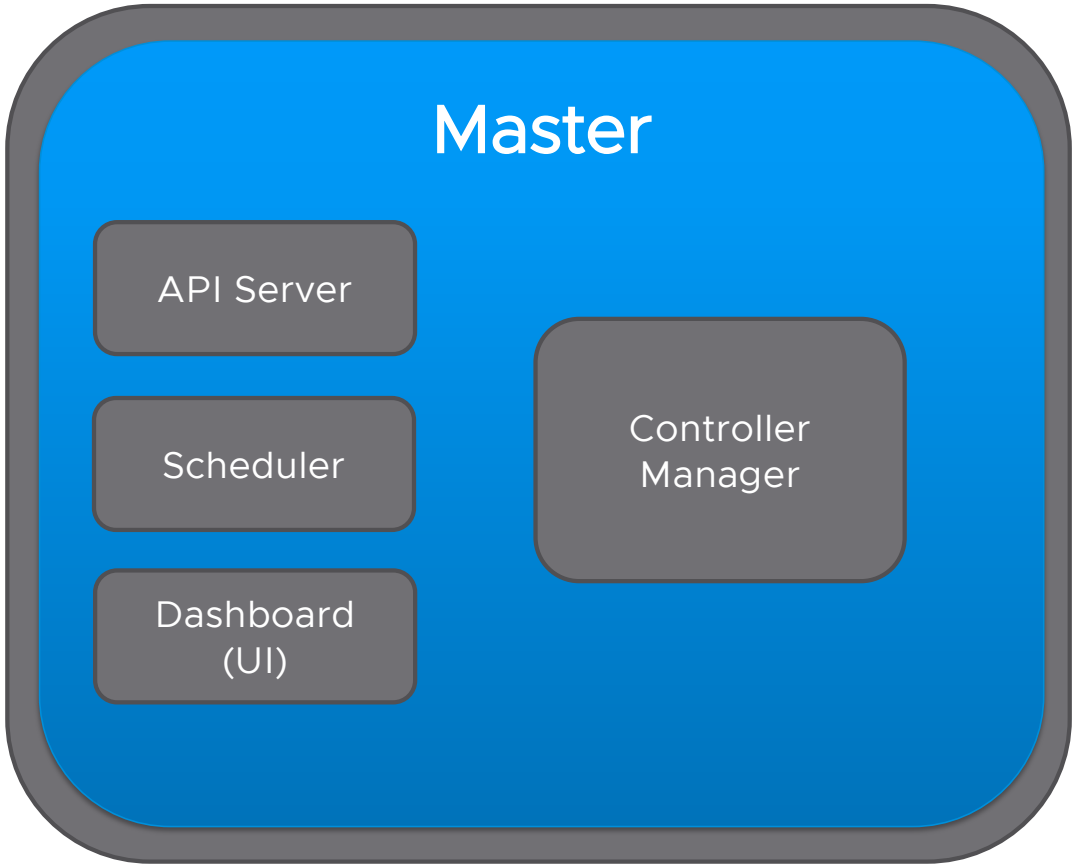
API Server

Scheduler

Dashboard (UI)

Controller Manager

- Replication Controller



K8s Worker Node

Kubelet

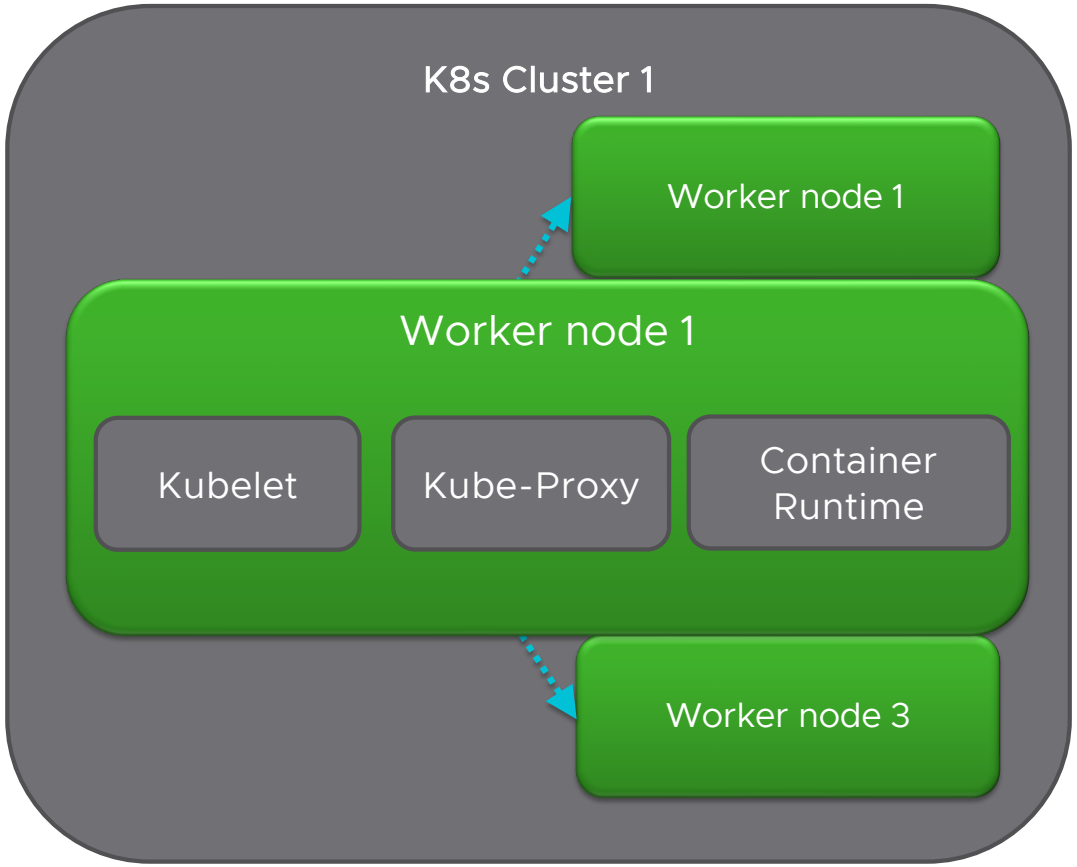
- Is an agent on the Nodes is watching for ‘PodSpecs’ to determine what it is supposed to run

Kube-Proxy

- Is a daemon watching the K8s ‘services’ on the API Server and implements east/west load-balancing on the nodes using NAT in IPTables

Container Runtime

- Docker
- Rkt



K8s etcd

Used as the distributed key-value store of Kubernetes

Stores configuration data that can be used by each of the nodes in the cluster

Can be distributed across multiple nodes

Used for service discovery

Represents the state of the cluster that each component can reference to configure or reconfigure themselves

Kubernetes Basic Concepts

vmworld®

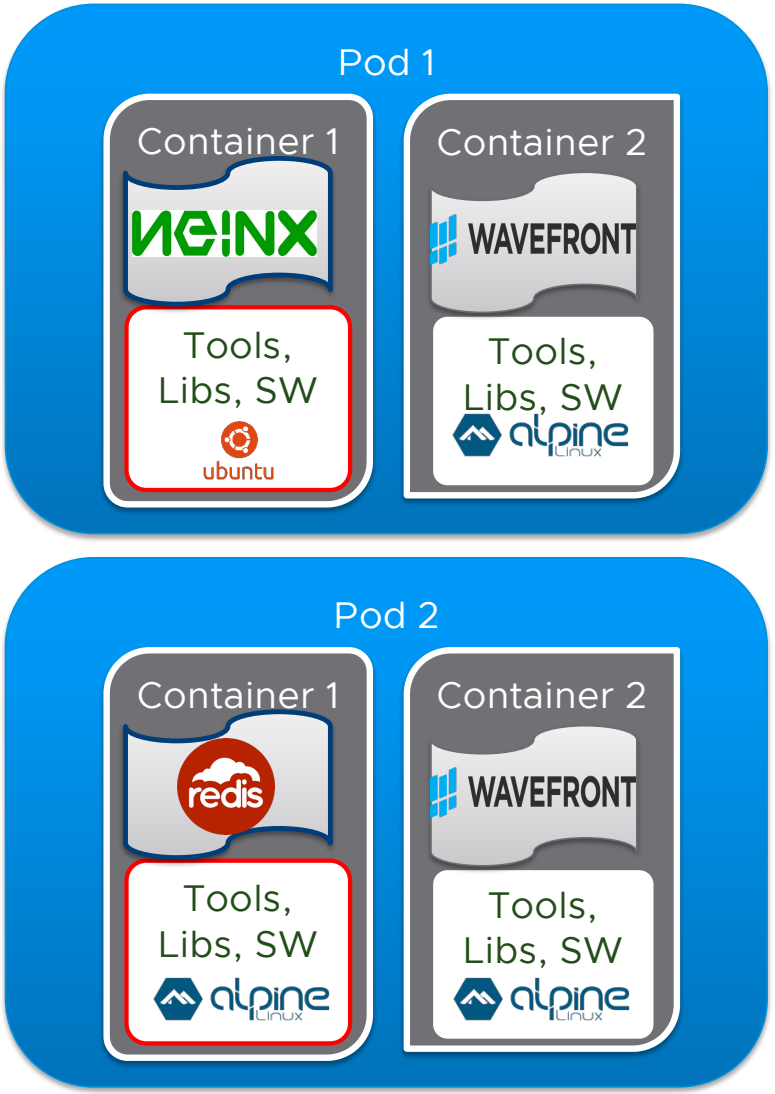
Pods

A pod (as in a pod of whales or pea pod) is a group of one or more containers

Containers within a pod share an IP address and port space, and can find each other via localhost

Containers in a Pod also share the same data volumes

Pods are considered to be ephemeral

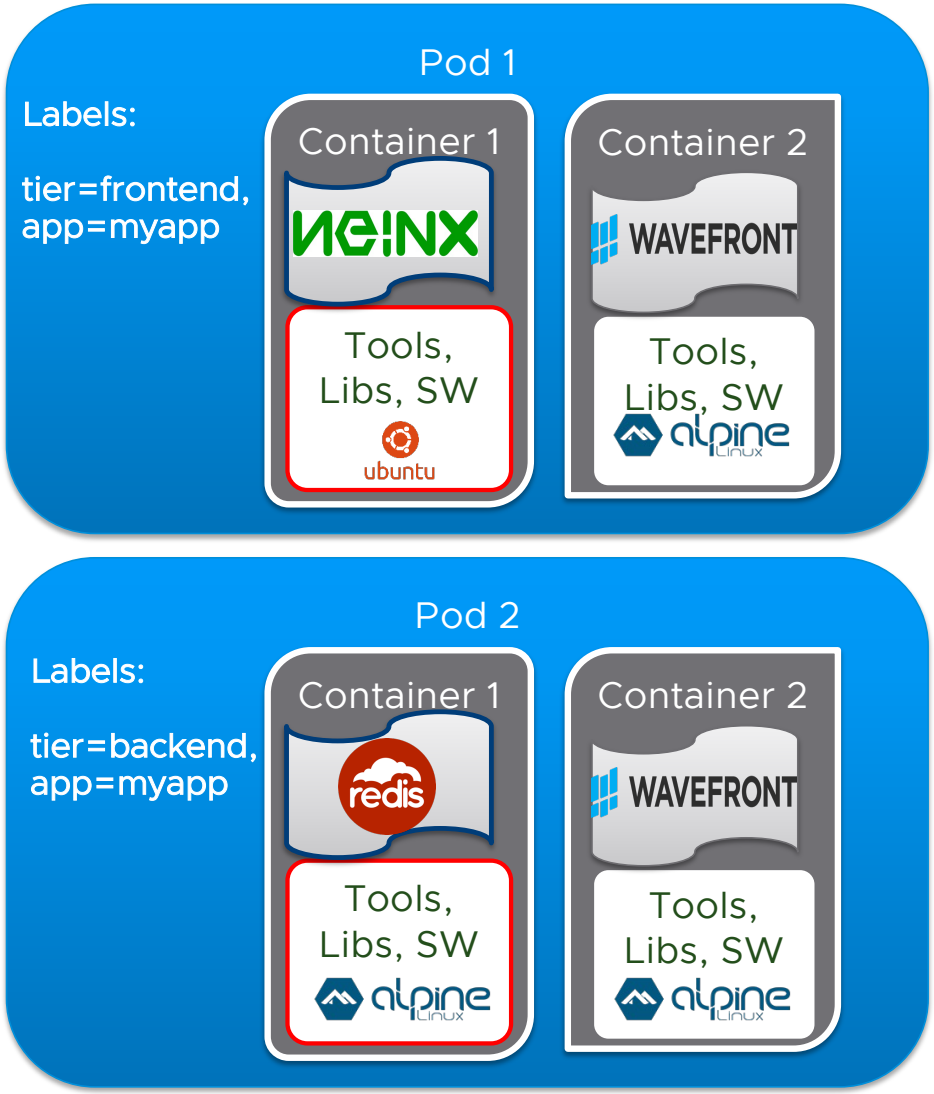


Labels

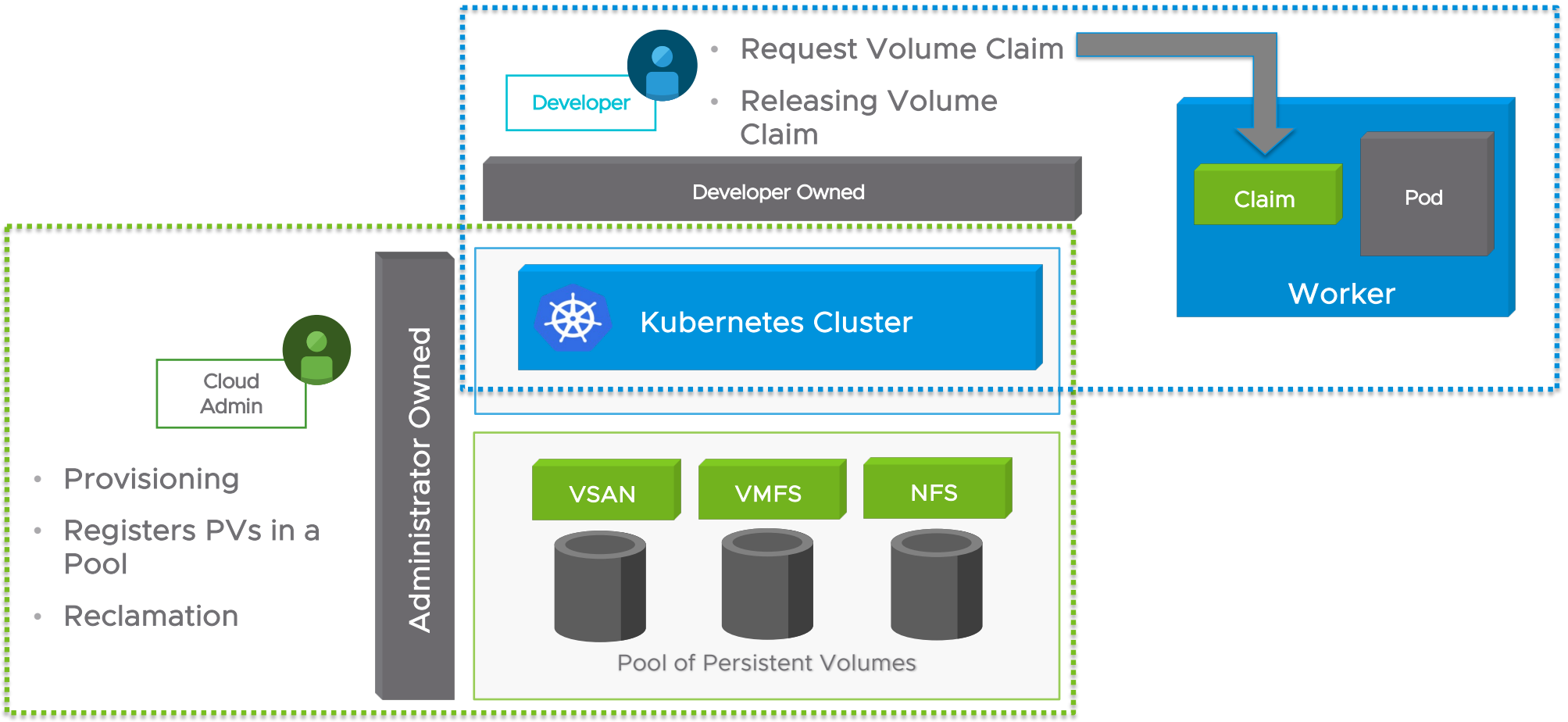
A Label is a key/value pair attached to Pods and convey user-defined attributes.

You can then use label selectors to select Pods with particular Labels and apply Services or Replication Controllers to them.

Labels can be attached to objects at creation time and subsequently added and modified at any time



Persistent Volume Claim



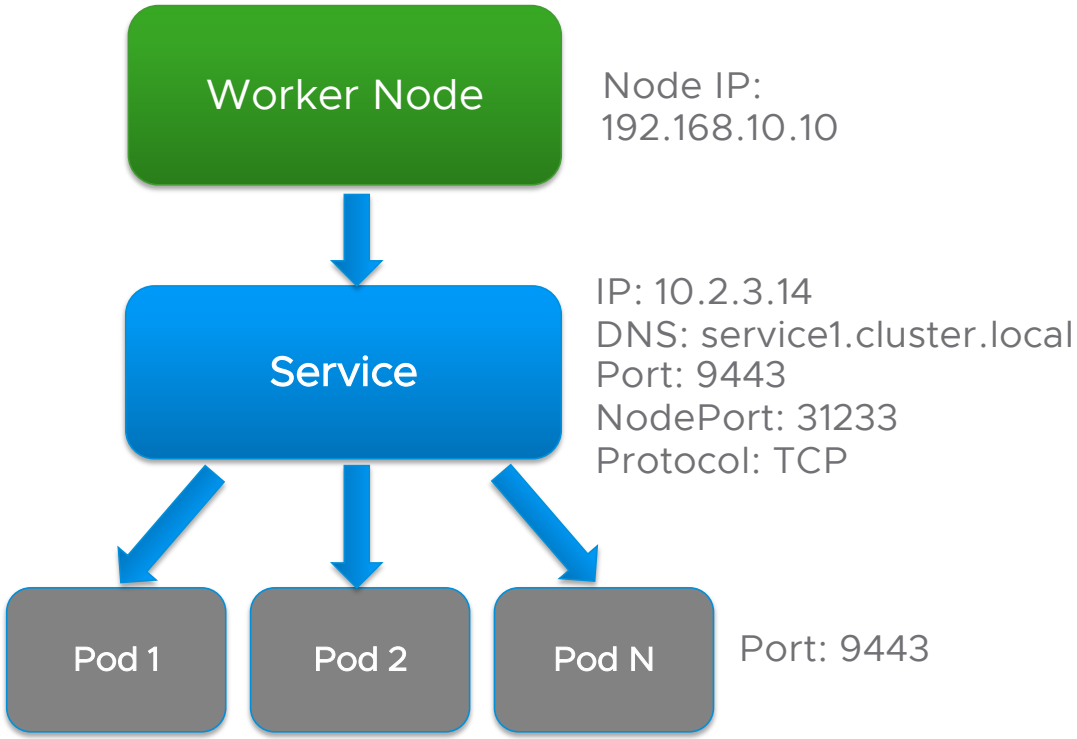
Services

Services Types

- ClusterIP
- NodePort
- Loadbalancer

Service Discovery

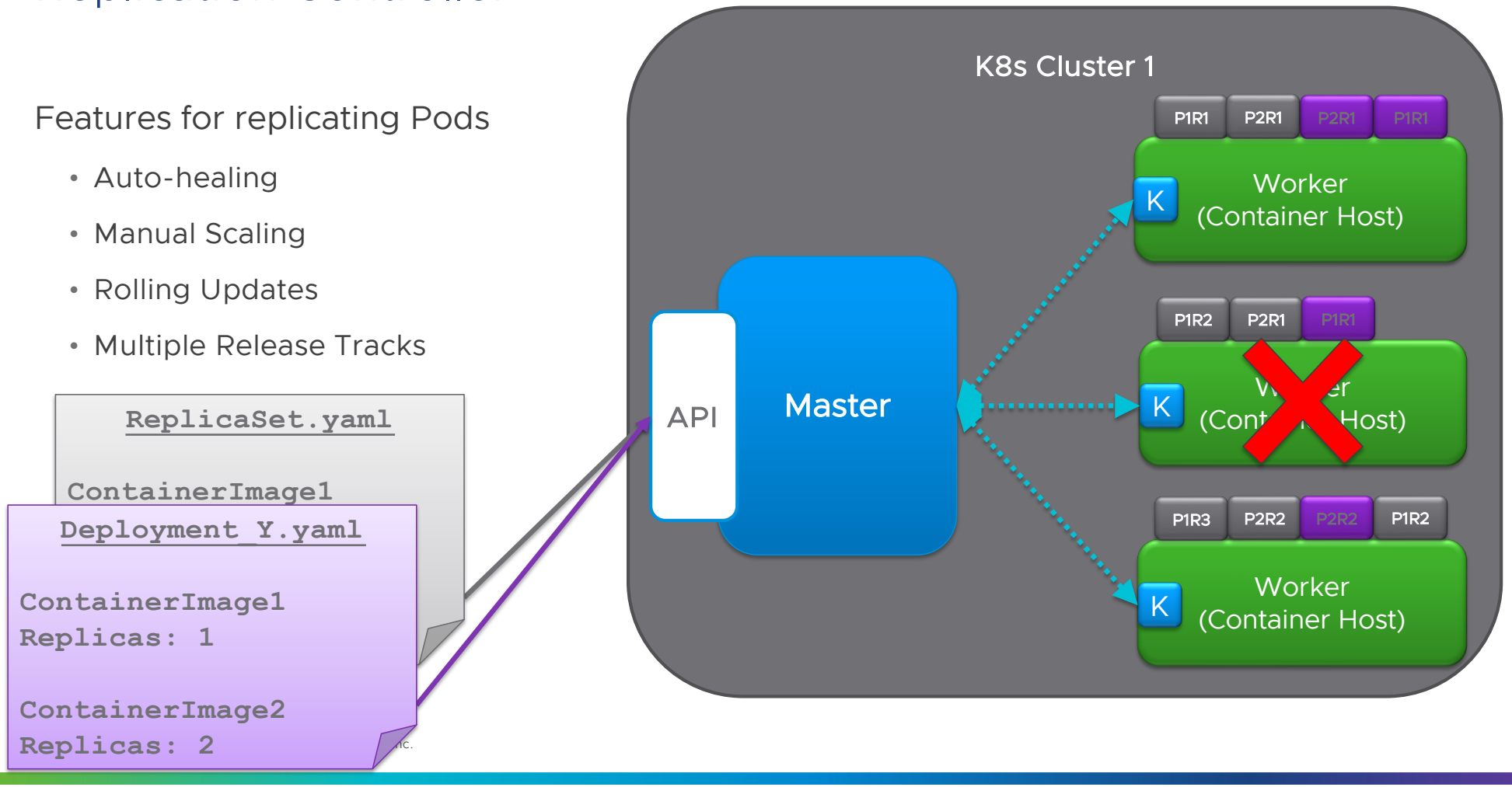
- DNS
- Environmental variables



Replication Controller

Features for replicating Pods

- Auto-healing
- Manual Scaling
- Rolling Updates
- Multiple Release Tracks



StatefulSet

The way of launching ordered replica's of Pods.

Enables running pods in "clustered mode"

- Master/Slave applications

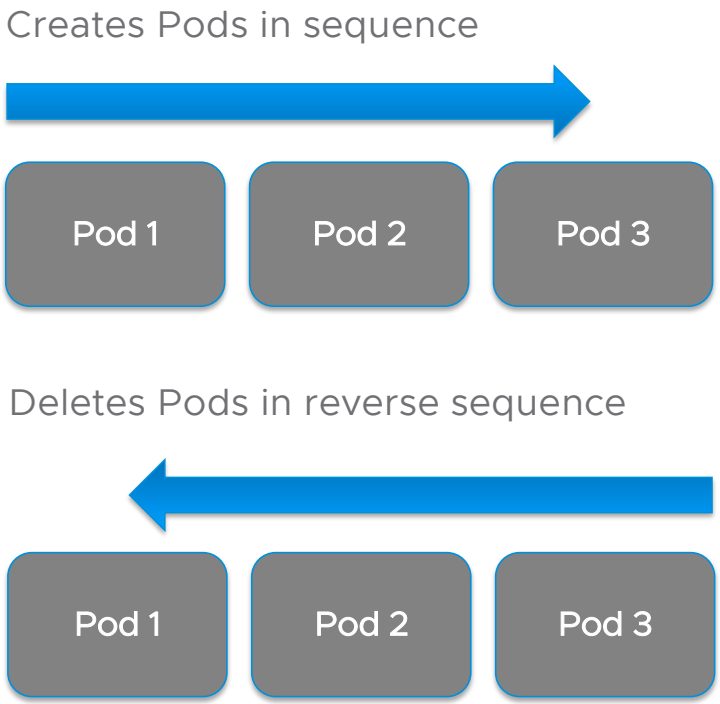
Valuable for applications that require:

- Stable and unique network identifiers
- Stable persistent storage
- Ordered deployment and scaling

Headless service required

Examples

- Zookeeper, Cassandra, etcd, MySQL, etc



DaemonSet

Runs a copy of a Pod on every node in the cluster

Newly created nodes automatically get the DaemonSet Pod(s)

When a node is removed a DaemonSet doesn't get rescheduled