



# Scheduling, Priority, and QoS

Ralph McNeal

Principal Engineer, CenturyLink Cloud & Portals

[ralphmcneal.com](http://ralphmcneal.com)

# Discussion Topics

Pod Eviction

Quality Of Service (QoS)

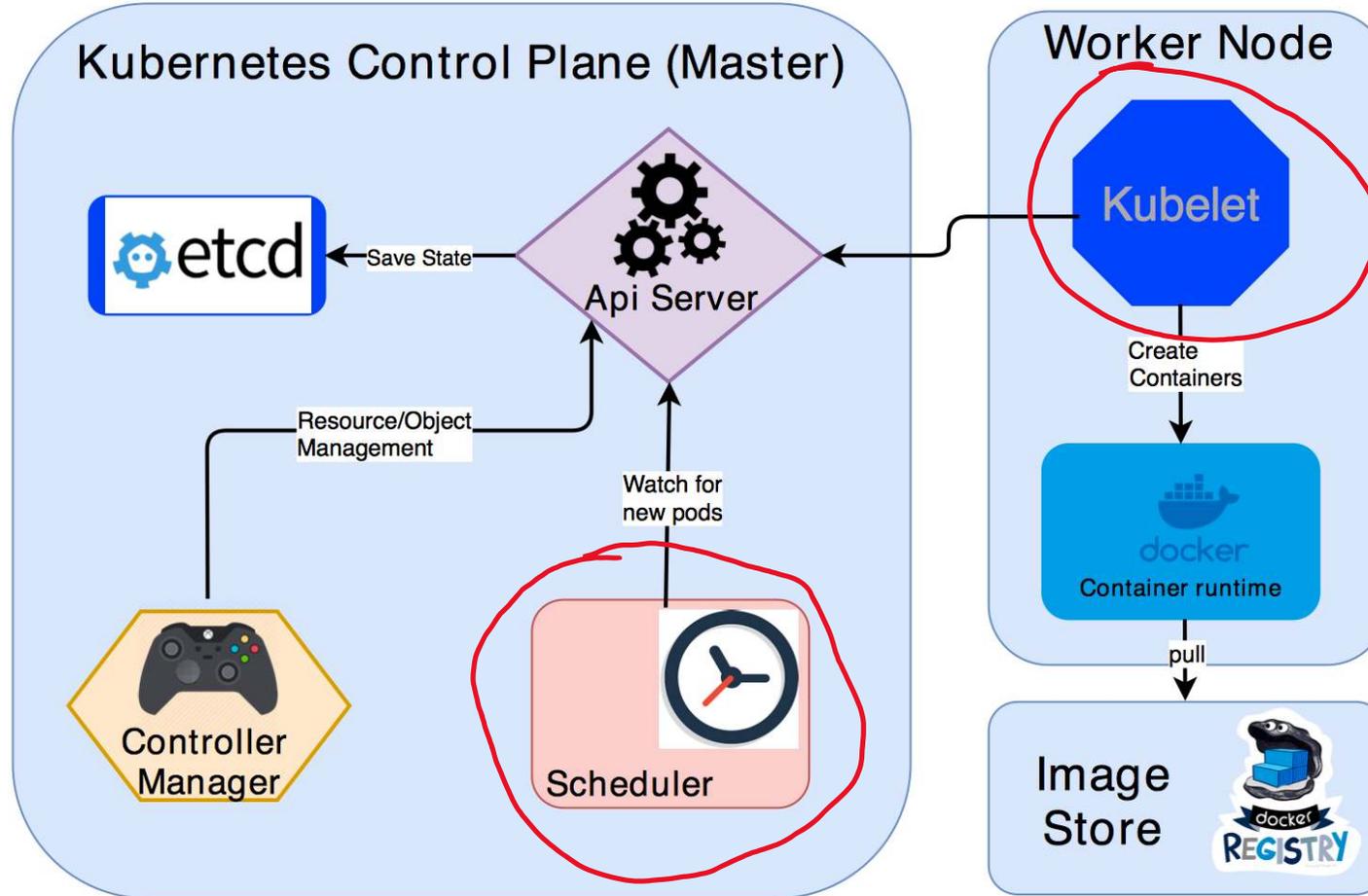
Managing Compute Resources

Pod Priority

Preemption

Pod Assignment

# Architecture Overview



# Pod Eviction



# Pod Eviction



Resource Eviction



Scheduling Eviction

# Quality of Service (QoS) Classes



Guaranteed



Burstable



Best-Effort



# QoS Level Assignment

given by Kubernetes

# Container Resource Management



Requests



Limits

# Container Resource Management

---

containers:

name: haproxy:2.0.4

resources:

**requests:**

cpu: 100m

memory: 1Gi

**limits:**

cpu: 100m

memory: 1Gi



**Guaranteed**

# Container Resource Management

---

containers:

name: clc-web-app:3.0.0

resources:

**requests:**

cpu: 100m

memory: 1Gi

**limits:**

cpu: 500m

memory: 2Gi



**Burstable**

# Container Resource Management

---

containers:

name: clc-poc:1.0.1



**Best-Effort**

# demo

---

Quality of Service (QoS)

# Container Resource Management



Scheduling?



Eviction?

Compressible  
Resources  
Vs.  
Incompressible  
Resources



Don't forget the **kubelet**

---

`--kube-reserved` and `--system-reserved`

# Determining Resource Specs

---

`docker stats command`

# Pod Priority

---

Priority indicates the importance of a Pod relative to other Pods.

# Pod Priority

---

```
apiVersion:  
kind: PriorityClass  
metadata:  
  name: high-priority  
value: 10000000  
globalDefault: false  
description: "For mission critical pods"
```

# Preemption

---

When a pod evicts a pod based on priority

# Non-preempting PriorityClass

---

apiVersion:

kind: PriorityClass

metadata:

  name: high-priority-nonpreempting

value: 10000000

**preemptionPolicy: Never**

globalDefault: false

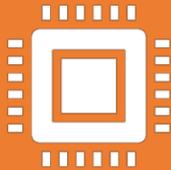
description: "This priority class will not cause other pods to be preempted."

# demo

---

Pod Priority and Preemption

# Interactions of Pod priority and QoS



The scheduler's preemption logic does not consider QoS



The kublet considers both QoS and Pod priority during [out-of-resource eviction](#)

# Pod Assignment



nodeSelector



Affinity

# Labels

---

- Key/value pairs on nodes, pods, and other objects
- Allows pod assignment via nodeSelector and affinity rules

# Node Selector

---

apiVersion: v1

kind: Pod

metadata:

  name: nginx

spec:

  containers:

    - name: nginx

      image: nginx

**nodeSelector:**

  disktype: ssd

# Affinity/Anti-Affinity

---

- More than just “and”
- Allows requirements to be “hard” or “soft”
- Allows constraints against labels on other pods vs. the node

# Node Affinity

---

affinity:

nodeAffinity:

requiredDuringSchedulingIgnoredDuringExecution:

nodeSelectorTerms:

- matchExpressions:

- key: disktype

operator: In

values:

- ssd

# Pod Affinity

---

affinity:

podAffinity:

requiredDuringSchedulingIgnoredDuringExecution:

- labelSelector:

matchExpressions:

- key: app

operator: In

values:

- cache

topologyKey: "kubernetes.io/hostname"

# Pod Anti-Affinity

---

affinity:

podAntiAffinity:

requiredDuringSchedulingIgnoredDuringExecution:

- labelSelector:

matchExpressions:

- key: app

operator: In

values:

- web

topologyKey: "kubernetes.io/hostname"

# Questions?

<http://kubernetes.io>  
<http://ralphmcneal.com>