

# **About Kubernetes RuntimeClass**

**- Let's select the container runtime as you like -**



**Makoto Hasegawa | @makocchi**  
**CyberAgent, Inc**

# About Me



Makoto Hasegawa

Hi

TWITTER / @makocchi  
FACEBOOK / makocchi0923

Adtech Division, CyberAgent, Inc

Infrastructure Engineer

Develop and maintain private OpenStack cloud

Develop and maintain Kubernetes as a Service platform on private cloud

Japan Container Days v18.12

「trunc だけじゃないコンテナ low level runtime 総合比較」

CKA (Certified Kubernetes Administrator) CKA-1700-0150-0100

CKAD (Certified Kubernetes Application Developer) CKAD-1800-0005-0100

# ***What is RuntimeClass?***

# ***What is RuntimeClass?***

# Runtime?

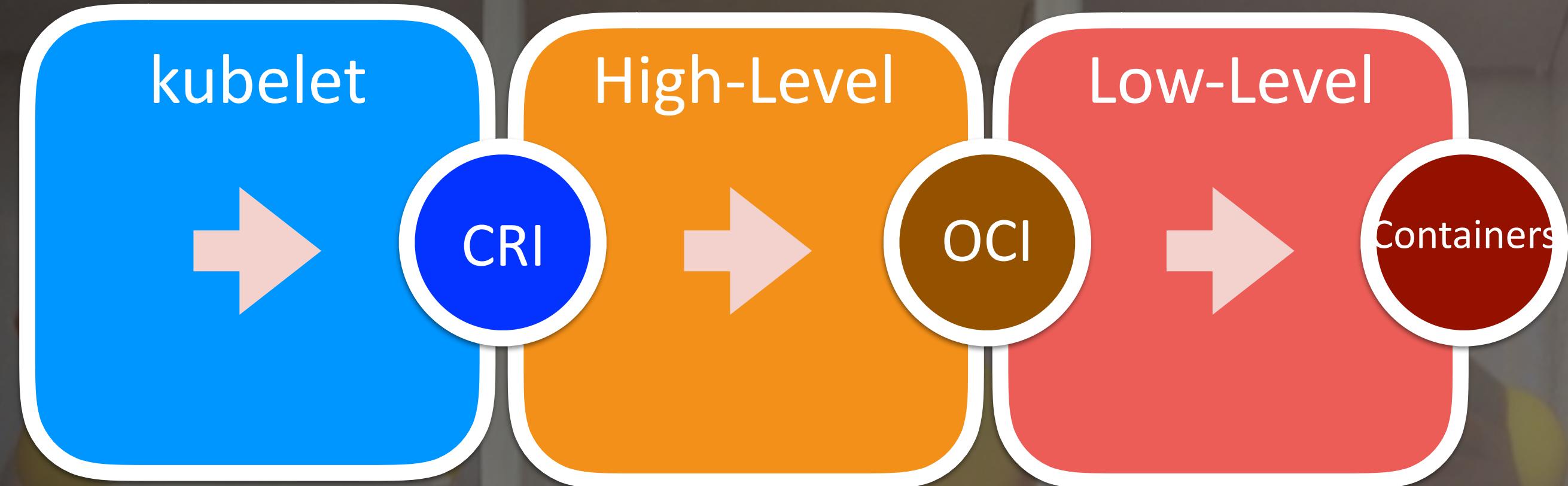
- ▶ Usually, the Runtime in the container world is defined two meanings by two layers.
- ▶ High-Level runtime(a.k.a CRI runtime)
  - ▶ Docker / Containerd / CRI-O ...
- ▶ Low-Level runtime(a.k.a OCI runtime)
  - ▶ runc / runsc / runnc / kata-runtime ...

# Runtime?

- ▶ Usually, the Runtime in the container world is defined two meanings by two layers.
- ▶ High-Level runtime(a.k.a CRI runtime)
  - ▶ Docker / Containerd / CRI-O ...
- ▶ Low-Level runtime(a.k.a OCI runtime)
  - ▶ runc / runsc / runnc / kata-runtime ...
- ▶ The Kubernetes RuntimeClass handles **Low-Level runtime**.

Runtimeclass

# Runtimes?



- ▶ Low-Level runtime(a.k.a OCI runtime)  
    RuntimeClass
- ▶ runc / runsc / runnc / kata-runtime ...
- ▶ The Kubernetes RuntimeClass handles **Low-Level runtime**.

# ***What is RuntimeClass?***

# RuntimeClass

- ▶ RuntimeClass is Kubernetes feature that makes Kubernetes user can select the low-level container runtime to run Pods.
- ▶ Kubernetes 1.12+ is required.
- ▶ RuntimeClass is **still alpha feature**  
(also Kubernetes 1.13.x)

WARNING

# *How to setup the RuntimeClass feature*

# How to Setup

## 1. Enable the RuntimeClass feature gate

Add “`--feature-gates=RuntimeClass=true`” to the `kube-apiserver` options and then restart the `kube-apiserver`.

## 2. Create the RuntimeClass CRD

`RuntimeClass` is provided by `CRD(Custom Resource Definition)` so you need to create the `CRD`.

```
$ kubectl apply -f https://raw.githubusercontent.com/kubernetes/kubernetes/master/staging/src/k8s.io/node-api/manifests/runtimeclass_crd.yaml  
customresourcedefinition.apiextensions.k8s.io/runtimeclasses.node.k8s.io configured
```

# How to Setup

## 3. Set up kubelet and container runtime on nodes

The RuntimeClass feature depends on CRI implementaion.  
If you are using Docker(dockershim) for High-Level  
container runtime for kubelet(this is default behavior),  
you need to change dockershim to containerd or CRI-O  
for the container runtime.

```
$ kubectl get node \
-o custom-columns=NAME:metadata.name,RUNTIME:.status.nodeInfo.containerRuntimeVersion
NAME      RUNTIME
node-1    docker://18.6.2
node-2    docker://18.6.2
node-3    cri-o://1.13.0
```

maybe not  
supported

yay! cool!!

# How to Setup

## 3. Set up kubelet and container runtime on nodes

Configure kubelet options to use CRI implementation.

### ▶ In case of CRI-O

--container-runtime=remote

--container-runtime-endpoint=unix:///var/run/crio/crio.sock

### ▶ In case of containerd

--container-runtime=remote

--container-runtime-endpoint=unix:///run/containerd/containerd.sock

# How to Setup

## 4. Create the RuntimeClass resources

This is example of using gVisor(runsc).

```
# runtimeclass.yaml
---
kind: RuntimeClass
apiVersion: node.k8s.io/v1alpha1
metadata:
  name: gvisor
spec:
  runtimeHandler: gvisor
```

```
$ kubectl apply -f runtimeclass.yaml
runtimeclass.node.k8s.io/gvisor created
```

# How to Setup

## *Further Reading*

### ▶ Install gVisor(runsc)

```
$ curl -L -s -o /usr/local/bin/runsc \
https://storage.googleapis.com/gvisor/releases/nightly/${YYYYMMDD}/runsc
```

```
$ chmod 755 /usr/local/bin/runsc
```

### ▶ Configure CRI-O to use gVisor

```
# Add following config to crio.conf
[crio.runtime.runtimes.gvisor]
runtime_path = "/usr/local/bin/runsc"
```

# *Create the Pod with RuntimeClass*

You need to specify “`.spec.runtimeClassName`” field in your Pod manifest.

```
# gvisor_pod.yaml
apiVersion: v1
kind: Pod
metadata:
  name: nginx
spec:
  runtimeClassName: gvisor
  containers:
  - name: nginx
    image: nginx
    imagePullPolicy: IfNotPresent
```

Insert  
here!

When you create Pods without `runtimeClassName`, Pods will run with default low-level runtime(depends on your high-level runtime configuration).

You can see a running pod with gvisor.

```
host $ ps -ef | grep nginx
root      10145      1  0 16:50 ?        00:00:00 /usr/local/libexec/crio/
common --syslog -c
67b935a32339c4c5be60beb5ff44e61fedebd52823e80a1cb187b1526eea3c73 -u
67b935a32339c4c5be60beb5ff44e61fedebd52823e80a1cb187b1526eea3c73 -r /usr/
local/bin/runsc -b /var/run/containers/storage/overlay-containers/
67b935a32339c4c5be60beb5ff44e61fedebd52823e80a1cb187b1526eea3c73/userdata
-p /var/run/containers/storage/overlay-containers/
67b935a32339c4c5be60beb5ff44e61fedebd52823e80a1cb187b1526eea3c73/
userdata/pidfile -l /var/log/pods/f05dacc3-042b-11e9-a981-fa2929175d9a/
nginx/0.log --exit-dir /var/run/crio/exits --socket-dir-path /var/run/
crio --log-level error
```

# Warning

If you set invalid value to runtimeClassName, the Pod will never reach the Running status.

Events:

Type	Reason	Age	From	Message
---	-----	----	-----	-----
Normal	Scheduled	75s	default-scheduler	
Successfully assigned default/nginx-hogehoge to node-1				
Warning	FailedCreatePodSandBox	8s (x6 over 74s)	kubelet, node-1	Failed create pod sandbox: runtimeclasses.node.k8s.io "hogehoge" not found

The Pod status will be ContainerCreating forever..

Create the Pod

# *The Future of* **the RuntimeClass**

# Migrate RuntimeClass from a CRD to an internal API

## #74433

Merged

k8s-ci-robot merged 4 commits into kubernetes:master from tallclair:runtimeclass-internal 10 days ago

Conversation 93

Commits 4

Checks 0

Files changed 151



tallclair commented 23 days ago • edited ▾

Member + 😊 ...

**What type of PR is this?**

/kind feature

**What this PR does / why we need it:**

Migrate the RuntimeClass API to a built-in Kubernetes API (from a CRD). For background, see this SIG-Architecture thread: <https://groups.google.com/d/topic/kubernetes-sig-architecture/RycGP4LvQZk/discussion>

**Which issue(s) this PR fixes:**

For [kubernetes/enhancements#585](#)

Future

The **RuntimeClass** will be built-in Kubernetes API from CRD. ([PR #74433](#))

And API version will change to “node.k8s.io/v1beta1” from “node.k8s.io/v1alpha1”

So you do not need to create the CRD for **RuntimeClass** with Kubernetes 1.14+.

Future



Let's select the container runtime as you like!!

# **THANK YOU !!**

## ***About Kubernetes RuntimeClass***

***- Let's select the container runtime as you like -***



***Makoto Hasegawa | @makocchi***

***CyberAgent, Inc***