

CRI Runtimes: Who  
is running my pod?



# Hello!

I'm Phil Estes



Distinguished Engineer & CTO,  
Linux OS & Container Architecture  
**IBM Watson & Cloud Platform**

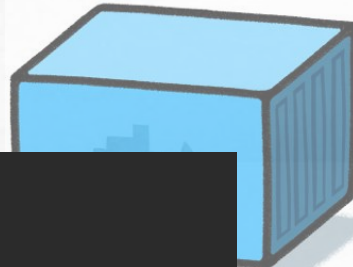
Docker Captain, containerd maintainer

[@estesp](#)

# Docker



# Containers



```
$ docker run redis  
$ docker ps  
$ docker stop redis  
$ docker build -t myapp .
```

You are using **Kubernetes** aren't you?



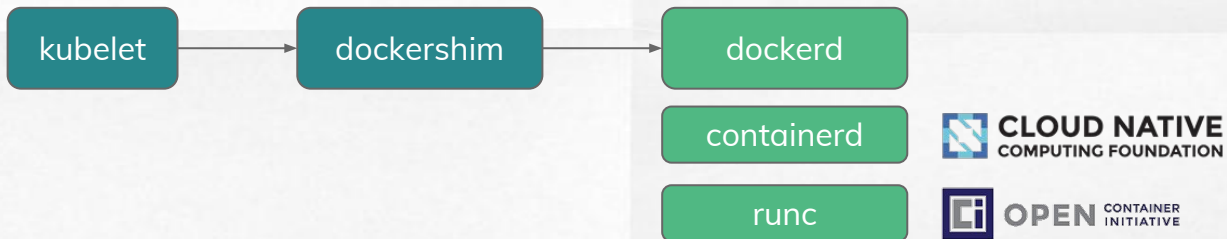
*(See the CNCF CCaC CoC)*

Kubernetes is an

orchestrator

# Kubernetes doesn't run your containers

<https://github.com/kubernetes/kubernetes/tree/release-1.4/pkg/kubelet/dockershim>



Monday, December 19, 2016

## Introducing Container Runtime Interface (CRI) in Kubernetes

*Editor's note: this post is part of a [series of in-depth articles](#) on what's new in Kubernetes 1.5*

At the lowest layers of a Kubernetes node is the software that, among other things, starts and stops containers. We call this the “Container Runtime”. The most widely known container runtime is Docker, but it is not alone in this space. In fact, the container runtime space has been rapidly evolving. As part of the effort to make Kubernetes more extensible, we've been working on a new plugin API for container runtimes in Kubernetes, called "CRI".

### What is the CRI and why does Kubernetes need it?

Each container runtime has its own strengths, and many users have asked for Kubernetes to support more runtimes. In the Kubernetes 1.5 release, we are proud to introduce the [Container Runtime Interface](#) (CRI) -- a plugin interface which enables kubelet to use a wide variety of container runtimes, without the need to recompile. CRI consists of a [protocol buffers](#) and [gRPC API](#), and [libraries](#), with additional specifications and tools under active development. CRI is being released as Alpha in [Kubernetes 1.5](#).

# Kubernetes

- ▣ K8s API
- ▣ Storage
- ▣ Networking (CNI)
- ▣ Healthchecks
- ▣ Placement
- ▣ Custom resources

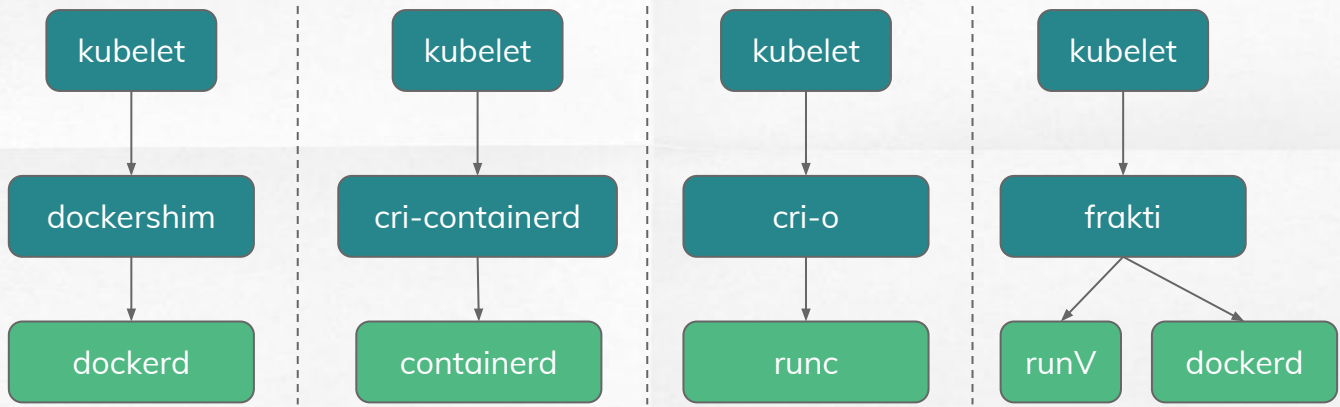
CRI

# Container Runtime

- ▣ Pod container lifecycle
  - Start/stop/delete
- ▣ Image management
  - Push/pull/status
- ▣ Status
- ▣ Container interactions
  - attach, exec, ports, log



# What Runtimes Exist?



```
kubelet --container-runtime {string}  
--container-runtime-endpoint {string}
```

But...

why should I  
care?



The benefits of runtime pluggability are mostly focused on operational concerns.




**Kelsey Hightower** 

@kelseyhightower

Following



*runtimes*

Treating containers  like a black box will eventually leave you in the dark.

10:52 AM - 22 Jun 2018

152 Retweets 614 Likes



 17

 152

 614



## What do I need?

- ▣ Performance
- ▣ Stability
- ▣ (Optional) Hypervisor Isolation
- ▣ Security Capabilities
- ▣ Broad Usage
- ▣ Multi-architecture Support

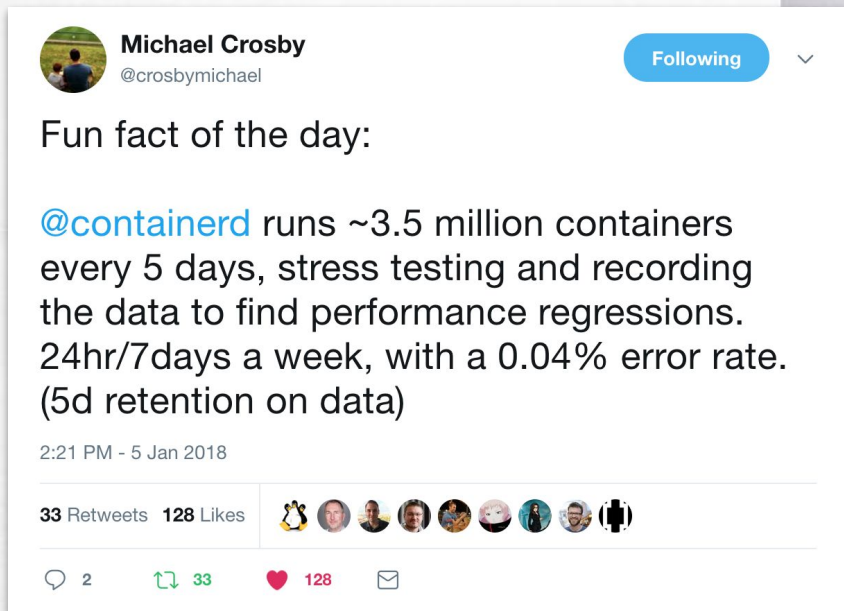
# Containerd: A Core Runtime


container 



## Containerd Benefits

- Designed with **broad usage** as a core container runtime:
  - Docker, LinuxKit, Kubernetes and embedded core runtime use cases (OpenWhisk, Cloud Foundry)
- **Stress testing** for stability and performance guarantees 24/7
- Usable Go library (or gRPC) for ease of embedding
- **Compatibility guarantees**; bug fix backports for stable support








 **Michael Crosby**  
@crosbymichael Following

Fun fact of the day:

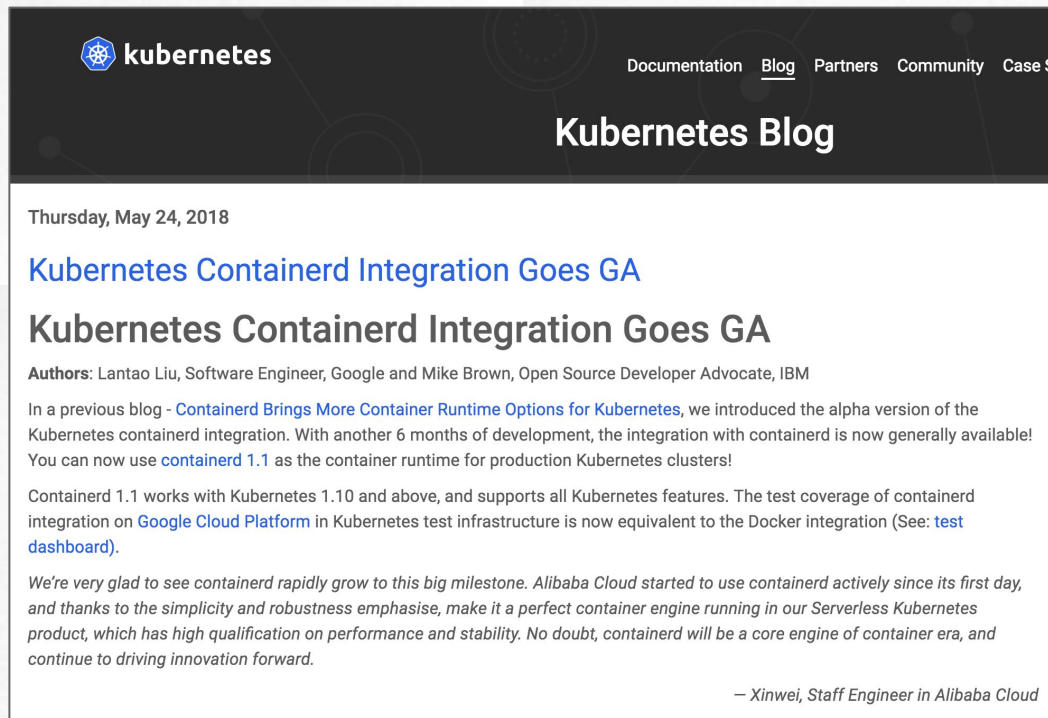
[@containerd](#) runs ~3.5 million containers every 5 days, stress testing and recording the data to find performance regressions. 24hr/7days a week, with a 0.04% error rate. (5d retention on data)

2:21 PM - 5 Jan 2018

33 Retweets 128 Likes 

 2  33  128 

# Containerd + CRI



The screenshot shows the top of a web browser displaying the Kubernetes Blog. The header is dark with the Kubernetes logo and name on the left, and navigation links for Documentation, Blog, Partners, Community, and Case Studies on the right. The main content area is white and features the date 'Thursday, May 24, 2018', the article title 'Kubernetes Containerd Integration Goes GA', and the author information: 'Authors: Lantao Liu, Software Engineer, Google and Mike Brown, Open Source Developer Advocate, IBM'. The article text begins with 'In a previous blog - Containerd Brings More Container Runtime Options for Kubernetes, we introduced the alpha version of the Kubernetes containerd integration. With another 6 months of development, the integration with containerd is now generally available! You can now use containerd 1.1 as the container runtime for production Kubernetes clusters!'. It continues with 'Containerd 1.1 works with Kubernetes 1.10 and above, and supports all Kubernetes features. The test coverage of containerd integration on Google Cloud Platform in Kubernetes test infrastructure is now equivalent to the Docker integration (See: test dashboard)'. A quote follows: 'We're very glad to see containerd rapidly grow to this big milestone. Alibaba Cloud started to use containerd actively since its first day, and thanks to the simplicity and robustness emphasise, make it a perfect container engine running in our Serverless Kubernetes product, which has high qualification on performance and stability. No doubt, containerd will be a core engine of container era, and continue to driving innovation forward.' The article concludes with the author's name: '— Xinwei, Staff Engineer in Alibaba Cloud'.

<https://kubernetes.io/blog/2018/05/24/kubernetes-containerd-integration-goes-ga/>



## Containerd in the Cloud(s)

- ❑ Kelsey Hightower's "[Kubernetes the Hard Way](#)" deploys **containerd** as the kubelet runtime
- ❑ **GKE alpha**: containerd-based K8s clusters
- ❑ **IBM Cloud**: containerd-based clusters in staging
- ❑ **Azure**: OSS acs-engine includes containerd; AKS moving to containerd (but CRI-O for OpenShift)
- ❑ **Amazon**: still reviewing runtime options for EKS
- ❑ **CloudFoundry**: moving to containerd from runc

# Kubernetes 1.11 + containerd 1.1.0

The screenshot shows the IBM Cloud Container Service dashboard for a Kubernetes cluster named 'containerd'. The top navigation bar includes 'IBM Cloud', 'Catalog', 'Docs', 'Support', and 'Manage', along with a user profile '(Philip...count)'. The breadcrumb path is 'Clusters / containerd'. The cluster status is 'Normal'. A 'Kubernetes Dashboard' link is available. The 'Overview' tab is selected, showing a 'Summary' table and a 'Worker Nodes' section. The 'Worker Nodes' section features a large green arc with the number '2' and the label 'Nodes', indicating 2 nodes in a normal state. A legend on the right shows 2 Normal (green), 0 Warning (yellow), 0 Critical (red), and 0 Pending (grey) nodes. A vertical 'FEEDBACK' button is on the right edge.

IBM Cloud Catalog Docs Support Manage (Philip...count)

Clusters / containerd

containerd ● Normal [Kubernetes Dashboard](#)

Access Overview Worker Nodes Worker Pools Services

Summary	
Cluster ID	c492459e4f4c400e9a3d988e8be9926e
Kubernetes version	1.11.0_1505
Zones	dal10
Owner	estesp@us.ibm.com
Infrastructure User	estesp
Ingress subdomain	containerd.us-south.stg.containers.appdomain.cloud
Metrics	<a href="#">View</a>

Worker Nodes

2 Nodes

- 2 ● Normal
- 0 ● Warning
- 0 ● Critical
- 0 ● Pending

FEEDBACK

Demo time

## Going Further

- ❑ **crictl** User's Guide:  
<https://github.com/containerd/cri/blob/master/docs/crictl.md>
- ❑ Stephen Day's **KubeCon 2018** containerd talk:  
<https://www.youtube.com/watch?v=3AynH3c0F8M>
- ❑ **Containerd** project:  
<https://github.com/containerd/containerd>

Thanks!

**Any questions?**

You can find me at:  
@estesp  
estesp@gmail.com

## Credits

---

Special thanks to all the people who made and released these awesome resources for free:

- ▣ Presentation template by SlidesCarnival
- ▣ Photographs by Unsplash
- ▣ Backgrounds by Pixeden