OCI-compatible haconiwa

hurdles and advantages
2019-04-12
RejectKaigi 2019 @ pixiv Inc

Yusuke Nakamura (unasuke)

about me

- Yusuke Nakamura (also known as "unasuke")
 - Employee of BANK Inc
 - Develop Rails application, manage Infrastructure <u>https://cash.jp/</u>
 - RubyKaigi 2019 helper
 - GitHub <u>@unasuke</u>
 - Twitter <u>@yu_suke1994</u>
 - Mastodon <u>@unasuke@mstdn.unasuke.com</u>



introduction

First, to clearly where we stand.

Your perception of containers

- Are you use container?
 - In production env? or(and) development env?
 - Use Docker? or the other one?
 - Orchestrate by ECS? or GKE? or on-premises?

We use Docker mostly

- de facto standard of a Linux container
 - Easy installation
 - for Mac, for Windows...
 - The first famous Linux container inplementation

"Container" is not equal "Docker"

- Before Docker
 - LXC (Linux)
 - Jail (FreeBSD)
 - etc...
- After Docker
 - cri-o
 - Kata Container
 - etc...

What's haconiwa

- The Linux contianer runtime written by C and mruby
 - <u>https://speakerdeck.com/udzura/the-alternative-</u> <u>container?slide=11</u>

OCIのspecを必ずしも満たすことは想定していな L)

- Independent from "Container" world
 - "Container" means OCI

What's OCI

The initialism of "Open Container Initiative"

https://www.opencontainers.org/

- OCI specs
 - Image spec
 - specifitation of the container image format
 - Runtime spec
 - specification of the container runtime interface

CRI and Kubernetes world



kubelet uses Container-Runtime-Interface(CRI) to communicate to container runtime

The kubelet is the primary "node agent" that runs on each node.

Diff of OCI/CRI compatible means...

- CRI compatible
 - usable as backend of kubelet
- OCI compatible
 - Exchangeable image and runtime

easy \rightarrow CRI compatible \rightarrow OCI compatible \rightarrow hard

Why CRI-compatible?

haconiwa is just run container. Doesn't orchestrate.

Pros

Orchestration by Kubernetes

Cons

Cannot use haconiwa-specific functions (hook)

maybe...

Why OCI-compatible?

- Pros
 - possible to share the existing assets
 - hub.docker.com
- Cons
 - Cannot use haconiwa-specific functions (hook)
 - https://github.com/haconiwa/haconiwa/blob/master/ sample/hooks.haco
 - maybe...

hurdles and advantages

- hurdles
 - it's hard to comply with the standard
- advantages
 - more users
 - wealth of existing assets

How to implement CRI

https://github.com/kubernetes/kubernetes/blob/ release-1.14/pkg/kubelet/apis/cri/runtime/v1alpha2/ api.proto

- Protocol Buffer
 - RuntimeService
 - ImageService
 - and many messages
- middleware?

CRI interface and haconiwa

- should start process to respond rpc
 - currently, haconiwa is just a command not service(or daemon)
- should implement rpc response interface

OCI specification and haconiwa

- image spec
 - should import/export OCI image
 - <u>https://blog.unasuke.com/2018/read-oci-image-spec-v101/</u>
- runtime spec
 - <u>https://udzura.hatenablog.jp/entry/</u> 2016/08/02/155913

conclusion

more resources, more users in OCI/CRI world

but...

- compliant to CRI is hard
- compliant to OCI is harder than CRI

conclusion



 \sim

めちゃくちゃ興味あり∅

Uchio KONDO 🥄 @udzura OCI対応したい、一緒にしてくれる方募集してます。当方ボーカル 🧸

13:06 - 2018年11月30日

https://twitter.com/yu_suke1994/status/ 1068355444928741376