KubeCon | CloudNativeCon
North America 2020
Virtual

CRI-O: The Runtime Control Room

Sascha Grunert (SUSE)

Urvashi Mohnani, Peter Hunt, Mrunal Patel (Red Hat)

Introduction

- What is CRI-O?
- Balance stability and new features

How CRI-O can be configured

Overview

- `/etc/crio/crio.conf` is the main configuration file written in TOML
- every configurable part of CRI-O can be set there:
    - specifying the `storage_driver` and `root`
    - which underlying OCI container runtime to choose (runc, crun for example)
    - security related options like the default `seccomp_profile`, `apparmor_profile` and used `default_capabilities`
    - debugging helpers like `log_level` and `log_filter`
- we're working on making configuration options dynamically reloadable
- introduced a modular configuration approach

How CRI-O can be configured

Dynamic aspects of CRI-O's configuration

- CRI-O supports a dynamic configuration reload
  - sending SIGHUP to the server reloads options which support the feature, like:
    - `seccomp_profile` and `apparmor_profile`
    - `log_level` and `log_filter`

- partial drop-in configurations can be stored in `/etc/crio/crio.conf.d`:

```
[crio.runtime]
log_level = "debug"
```

- alphabetical order of processing of the snippets make them easy to use
- works with the dynamic configuration reload feature, too

How CRI-O can be configured

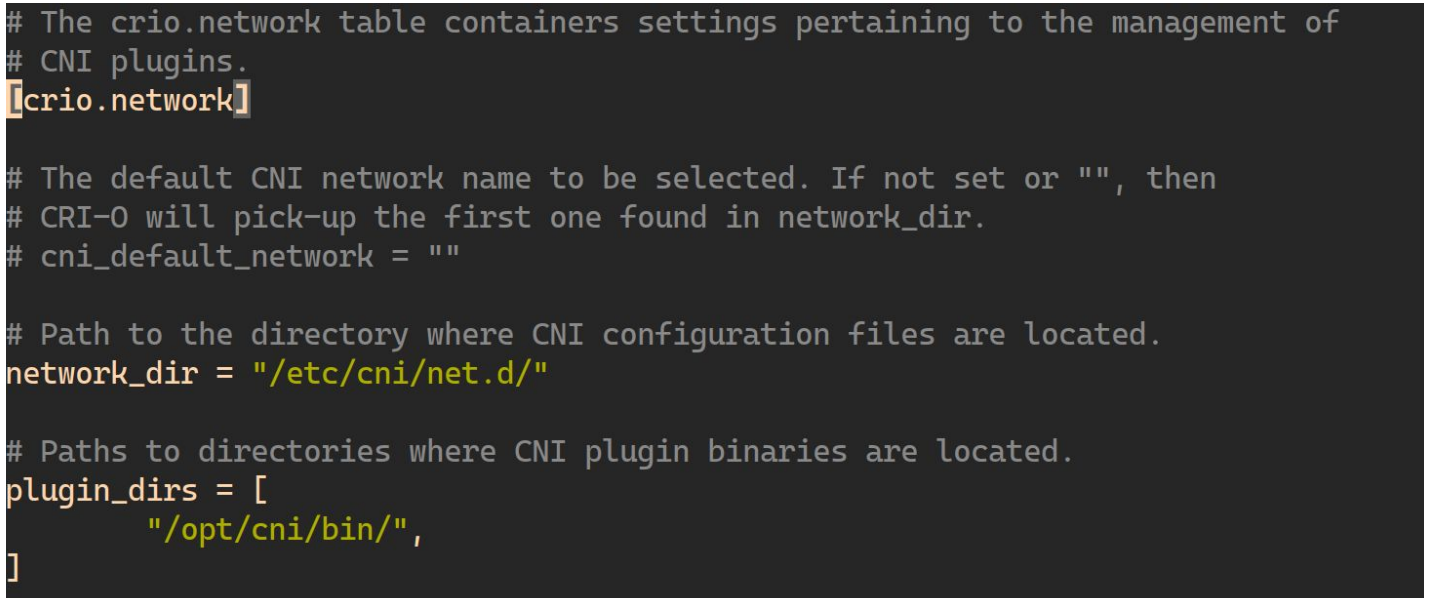
Shared configurations between the container tools

- Shared backend libraries
  - containers/storage
  - containers/image
- `/etc/containers/registries.conf`
  - Configure insecure, blocked, unqualified-search registries, and mirrors
- `/etc/containers/policy.json`
  - Policy requirements for a container image
- `/etc/containers/storage.conf`
  - Configure various storage related options such driver, runroot, size etc.

How CRI-O can be configured

Networking configuration

- CRI-O uses CNI for configuring networking for k8s pods
- Any CNI compatible plugins are supported
- CRI-O allows bootstrapping networking through daemonsets
- CRI-O supports setting a default networking plugin

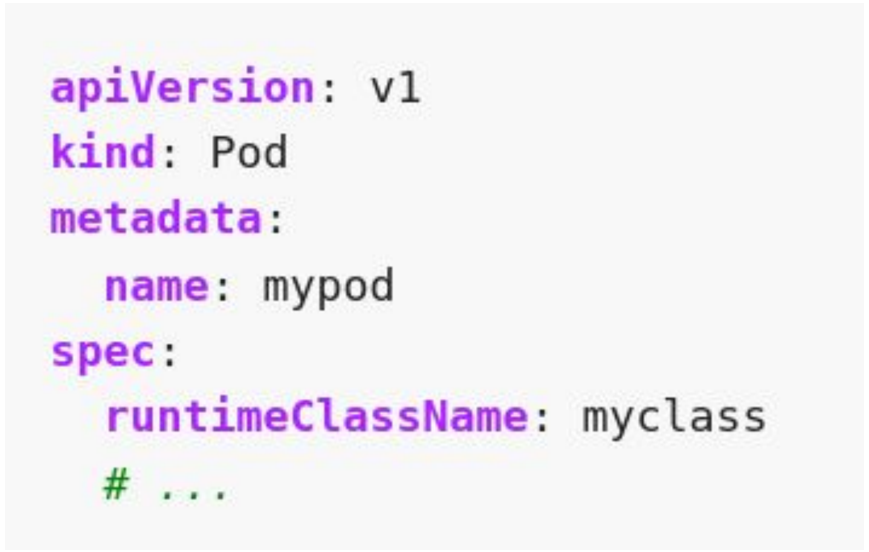


```
# The crio.network table containers settings pertaining to the management of
# CNI plugins.
[crio.network]

# The default CNI network name to be selected. If not set or "", then
# CRI-O will pick-up the first one found in network_dir.
# cni_default_network = ""

# Path to the directory where CNI configuration files are located.
network_dir = "/etc/cni/net.d/"

# Paths to directories where CNI plugin binaries are located.
plugin_dirs = [
        "/opt/cni/bin/",
]
```

How CRI-O can be configured

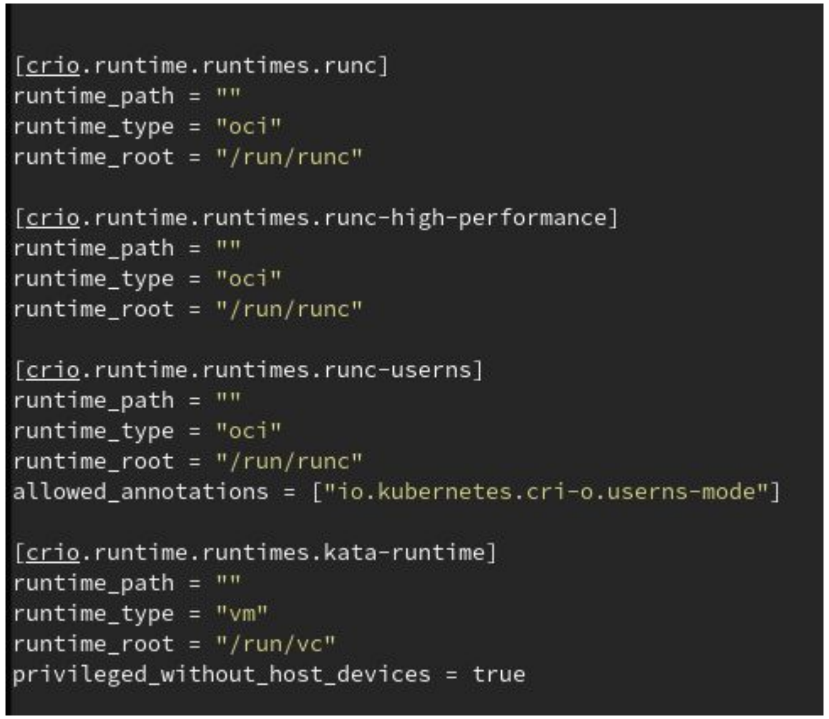
Runtime Classes/Handlers

- Differing workloads have different performance/security needs
- Runtime Classes (GA in 1.20) asks CRI implementation to use a different runtime
  - Or use it differently
- Admins can create runtime classes, and add admission controllers/policies to gate them



```yaml
apiVersion: v1
kind: Pod
metadata:
  name: mypod
spec:
  runtimeClassName: myclass
  # ...
```

https://kubernetes.io/docs/concepts/containers/runtime-class/



```
[crio.runtime.runtimes.runc]
runtime_path = ""
runtime_type = "oci"
runtime_root = "/run/runc"

[crio.runtime.runtimes.runc-high-performance]
runtime_path = ""
runtime_type = "oci"
runtime_root = "/run/runc"

[crio.runtime.runtimes.runc-userns]
runtime_path = ""
runtime_type = "oci"
runtime_root = "/run/runc"
allowed_annotations = ["io.kubernetes.cri-o.userns-mode"]

[crio.runtime.runtimes.kata-runtime]
runtime_path = ""
runtime_type = "vm"
runtime_root = "/run/vc"
privileged_without_host_devices = true
```

How CRI-O can be configured

Pod Annotations

- Key/Value map in Pod metadata
- Allow passing of unstructured data to varying levels of the stack
- CRI-O specific annotations:
    - UsernsModeAnnotation: "io.kubernetes.cri-o.userns-mode"
    - ShmSizeAnnotation: "io.kubernetes.cri-o.ShmSize"

How CRI-O can be configured

KubeCon CloudNativeCon North America 2020 Virtual

Summary

- Admins can:
  - Configure CRI-O specific configuration
  - Add runtime classes to restrict varying behavior
  - Have admission controllers gate runtime classes/annotations before they reach CRI-O

https://star-wars-memes.fandom.com/wiki/Unlimited_power!?file=CFD5C99C-B322-46A2-9892-AFDF484D28E1.jpeg

Runtime Class Topology

Manage CPU load balancing for workloads

- Enable/disable CPU load balancing
  - High performance runtimes
  - `cpu-load-balancing.crio.io` annotation set in pod/container spec
  - Hooks for pre-start and pre-stop run

Security related configurations

seccomp profile override

Kubernetes Container Runtime Interface (CRI) defines the main behavior, which is not always the most secure.

- not specified seccomp profiles are right now considered as `unconfined`:
  ```
  // Default: "", which is identical with unconfined.
  string seccomp_profile_path = 7;
  ```

- new configuration option `seccomp_use_default_when_empty` will help to increase the security defaults
- turning the option on will apply the `runtime/default` seccomp profile to all workloads which do not explicitly specify `unconfined` or a `localhost/` profile

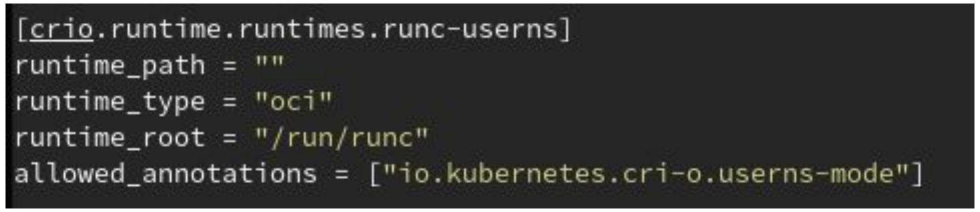
User namespaces

- users/groups = people
- available range of IDs = house
- user namespace = people living in a doll house (subset of range of IDs)
- Security advantage:
  - While inside the container, the process thinks it is privileged
  - Outside, it can be an unprivileged process

https://live.staticflickr.com/2772/4426922434_fba42eb481_w_d.jpg

User namespaces

- Long time issue:
  - https://github.com/kubernetes/enhancements/issues/127 (2016)
  - https://github.com/kubernetes/enhancements/pull/2101 (2020)
- Much like pids_limit, CRI-O has added support before upstream kube
- Admins can
  - stop anyone from creating user namespaces
  - only allow some to use user namespaces (admission controller/policy/runtime class)
  - give anyone access to user namespaces



```
[crio.runtime.runtimes.runc-userns]
runtime_path = ""
runtime_type = "oci"
runtime_root = "/run/runc"
allowed_annotations = ["io.kubernetes.cri-o.userns-mode"]
```

Future

Currently at CRI-O 1.19

- Cgroups V2
- User namespaces
- Using Rust for some of the cri-o components
- Graduation!

Find out More!

Website	https://cri-o.io
Github	https://github.com/cri-o/cri-o
Slack	#crio on kubernetes.slack.com
IRC	#crio on Freenode
Past Talks	https://github.com/cri-o/cri-o/blob/master/awesome.md
Coloring Book	https://github.com/mairin/coloringbook-container-commandos/blob/master/Web.pdf

KEEP CLOUD NATIVE EVERYWHERE

KubeCon | CloudNativeCon

North America 2020

Virtual