send in the `chowns`

systemd containers on OpenShift

Fraser Tweedale

`@hackuador`

Red Hat

January 15, 2022

Preliminaries

- CC-BY 4.0, except where otherwise noted
- Slides are available at speakerdeck.com/frasertweedale
- I will be available in the chatroom following the presentation

Agenda

- Containers and container standards
- Kubernetes and OpenShift
- FreeIPA: overview and use cases
- FreeIPA and systemd-based workloads on Kubernetes/OpenShift
    - challenges, workarounds, solutions

What is a container?

- An process isolation and confinement *abstraction*
- Most commonly: OS-level virtualisation (shared kernel)
  - e.g. FreeBSD jails, Solaris zones
- Container ***image*** defines filesystem contents

Containers on linux

- namespaces: pid, mount, network, cgroup, ...
- (maybe) SELinux/AppArmor
- (maybe) restricted capabilities(7) or seccomp(2) profile

Container standards

- *Open Container Initiative (OCI)*[1]
- **Runtime Specification**[2] - low level runtime interface
  - Linux, Solaris, Windows, VMs, . . .
  - Implementations[3]: **runc**[4] (reference implementation), crun[5], Kata Containers[6]

[1] https://opencontainers.org

[2] https://github.com/opencontainers/runtime-spec

[3] https://github.com/opencontainers/runtime-spec/blob/main/implementations.md

[4] https://github.com/opencontainers/runc

[5] https://github.com/containers/crun

[6] https://katacontainers.io/

OCI Runtime Specification

- JSON configuration (example[7])
- mounts, process and environment, lifecycle hooks, . . .
- Linux-specific: capabilities, namespaces, cgroup, sysctls, `seccomp` profile

[7] https://github.com/opencontainers/runtime-spec/blob/main/config.md#configuration-schema-example

Kubernetes and OpenShift

Kubernetes - container orchestration

- Abbreviation: **“k8s”**
- A container orchestration system
- Declarative configuration of container-based applications
- Integration with many cloud providers
- https://kubernetes.io/
- https://github.com/kubernetes/

Kubernetes - terminology

- **Container**: isolated/confined process [tree]
- **Pod**: group (1+) of related Containers (e.g. HTTP app + database)
- **Namespace**: object and auth[nz] scope, such as for a team/project
- **Node**: a machine in the cluster; where Pods are executed

Kubernetes - more terminology

- **Kubelet**[8]: agent that executes Pods on Nodes
- **Sandbox**: isolation/confinement mechanism(s); one per Pod
- **Container Runtime Interface (CRI)**[9]: interface used by Kubelet to create/start/stop/destroy Sandboxes and Containers
  - CRI-O[10]
  - containerd[11]

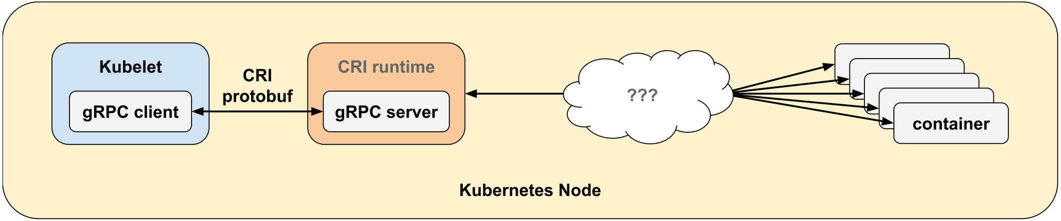
[8] https://kubernetes.io/docs/reference/command-line-tools-reference/kubelet/

[9] https://kubernetes.io/docs/concepts/architecture/cri/

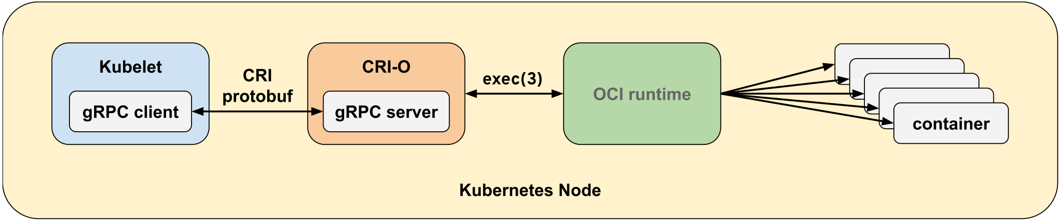
[10] https://cri-o.io/

[11] https://containerd.io/

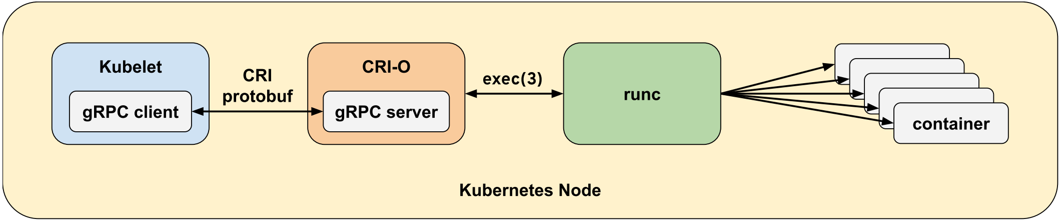
Kubernetes - Container Runtime Interface



Kubernetes - Container Runtime Interface - CRI-O



Kubernetes - Container Runtime Interface - CRI-O + runc



Kubernetes - Pod definition

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: fedora
  labels:
    app: fedora
spec:
  containers:
  - name: fedora
    image: registry.fedoraproject.org/fedora:35-x86_64
    command: ["sleep", "3600"]
    env:
    - name: DEBUG
      value: "1"
```

OpenShift[13]

- a.k.a. *OpenShift Container Platform (OCP)*
- An *enterprise-ready Kubernetes container platform*
- Commercially supported by Red Hat
- Community “upstream” distribution: OKD[12]
- Uses **CRI-O** and **runc**
- Latest stable release: 4.9

[12] https://www.okd.io/

[13] https://openshift.com/

OpenShift - terminology

- All existing Kubernetes terminology, plus...
- **Project**: Extends the *Namespace* concept
- **Security Context Constraint (SCC)**: policy affecting SELinux context, `seccomp` profile, `capabilities`, UID

OpenShift runtime environment (today)

- Sandboxes use SELinux, namespaces (cgroup, pid, mount, uts, network)
- Each *Project* gets assigned a unique UID range
- Containers run as a UID from that range
  - Circumvent via `RunAsUser` and SCCs (**bad idea**)

FreeIPA

FreeIPA

- Open Source identity management solution
- Users, groups, services, authentication, access policies
- 389 DS (LDAP), MIT Kerberos, Apache, Dogtag PKI, SSSD, . . .
- Part of RHEL (commercial support) and Fedora (community support)
- https://www.freeipa.org/

FreeIPA on Kubernetes/OpenShift - use cases

Identity services. . .

- for business applications running on the cluster
- for the cluster itself (API access, node access)
- for an entire organisation, hosted on their OpenShift cluster
- *as a service*, hosted and managed by a service provider

FreeIPA container

- Encapsulate the whole RHEL/Fedora-based system in a container
- PID 1 is `systemd`, which starts/manages all services
- We call this a *monolithic container*

Whyyyy?!

- Big engineering effort to rearchitect FreeIPA to be "cloud native"
- *Ongoing costs* as we support two different application architectures
- If we were starting from scratch today. . .

FreeIPA on OpenShift - challenges

- Unsurprisingly, there are many
- Main areas:
  - **runtime**
  - volumes and mounts
  - ingress[14,15]

[14] https://frasertweedale.github.io/blog-redhat/posts/2021-11-18-k8s-tcp-udp-ingress.html

[15] https://frasertweedale.github.io/blog-redhat/posts/2020-12-08-k8s-srv-limitation.html

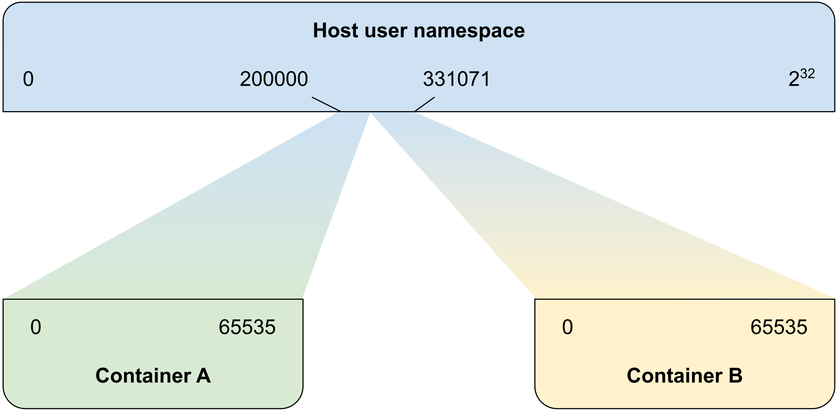
Challenges, workarounds and solutions

Runtime - user namespaces

- systemd and other components expect to run as `root` or other specific UID
- Solution: `user_namespaces(7)`
  - Implemented in CRI-O, since OpenShift 4.7
  - Opt-in via Pod annotation
  - Requires non-default cluster configuration
  - Requires Pod to be admitted via `anyuid` (or similar) SCC[16]

[16] I am working on a way to avoid this

Runtime - user namespaces



Runtime - user namespaces

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: nginx
  labels:
    app: nginx
  annotations:
    io.openshift.builder: "true"
    io.kubernetes.cri-o.userns-mode: "auto:size=65536"
spec:
  containers:
  - name: nginx
    image: quay.io/ftweedal/test-nginx:latest
    tty: true
```

Runtime - user namespaces - Kubernetes support

- KEP[17]-127: a long-running and ongoing discussion
- First proposal: https://github.com/kubernetes/enhancements/pull/1903
- Second proposal: https://github.com/kubernetes/enhancements/pull/2101
- Current proposal: https://github.com/kubernetes/enhancements/pull/3065

[17] Kubernetes Enhancement Proposal

Runtime - cgroups

- OpenShift creates a unique cgroup[18] for each container
- cgroup namespace[19] makes it the “root” namespace inside the container
- cgroupfs mounts it at `/sys/fs/cgroup`
- `systemd` needs write access… but doesn’t have it

[18] `cgroups(7)`

[19] `cgroup_namespaces(7)`

Runtime - cgroup ownership

- Solution: modify runtime to `chown` the cgroup to the container process UID
- But first: extend OCI Runtime Spec with semantics for cgroup ownership[20]
- runc pull request[21]
  - Merged; release expected in OpenShift 4.11 or later

[20] https://github.com/opencontainers/runtime-spec/blob/main/config-linux.md#cgroup-ownership

[21] https://github.com/opencontainers/runc/pull/3057

Runtime - OCI cgroup ownership semantics

`chown` container's cgroup to host UID matching the process UID in container's user namespace, if and only if...

- cgroups v2 in use, and
- container has its own cgroup namespace, and
- cgroupfs is mounted read/write

Runtime - OCI cgroup ownership semantics

Only the cgroup directory itself, and the files mentioned in `/sys/kernel/cgroup/delegate`, should be `chown`'d:

- `cgroup.procs`
- `cgroup.threads`
- `cgroup.subtree_control`
- `memory.oom.group`[22]

[22] depends on kernel version

Runtime - cgroups v2

- cgroups v2 is required for secure cgroup delegation
- it works, but is not yet the default cluster configuration
- it is on the roadmap

Runtime - cluster configuration (OCP 4.10) - 1/3

```
apiVersion: machineconfiguration.openshift.io/v1
kind: MachineConfig
metadata:
  name: enable-cgroupv2-workers
  labels:
    machineconfiguration.openshift.io/role: worker
spec:
  kernelArguments:
    - systemd.unified_cgroup_hierarchy=1
    - cgroup_no_v1="all"
    - psi=1
  ...
```

Runtime - cluster configuration (OCP 4.10) - 2/3

```
config:
  ignition:
    version: 3.1.0
  storage:
    files:
    - path: /etc/subuid
      overwrite: true
      contents:
        source: data:text/plain;charset=utf-8;base64,Y29
    - path: /etc/subgid
      overwrite: true
      contents:
        source: data:text/plain;charset=utf-8;base64,Y29
  ...
```

Runtime - cluster configuration (OCP 4.10) - 3/3

```
systemd:
  units:
  - name: "rpm-overrides.service"
    enabled: true
    contents: |
      [Unit]
      Description=Install RPM overrides
      After=network-online.target rpm-ostreed.service
      [Service]
      ExecStart=/bin/sh -c 'rpm -q runc-1.0.3-992.rhac
        || rpm-ostree override replace --reboot https://f
      Restart=on-failure
      [Install]
      WantedBy=multi-user.target
```

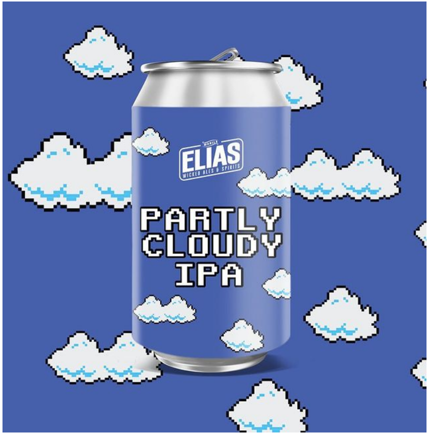
Demo

Links / resources

- Project main repo: https://github.com/freeipa/freeipa-openshift
  - not much here yet, watch this space
- runc builds: https://ftweedal.fedorapeople.org/
- Team blogs:
  - https://frasertweedale.github.io/blog-redhat/tags/containers.html
  - https://avisiedo.github.io/docs/
- Demo: https://www.youtube.com/watch?v=OGAVvIJwmd0

Status and future

- Kubernetes: user namespaces support in an ongoing discussion
- OpenShift: systemd container in user namespace works, but experimental
- Official support is an open question
  - We are hopeful, collaborating with OpenShift project and product management, looking for allies
  - But we may end up having to rearchitect FreeIPA for the cloud



Elias Wicked Ales & Spirits
https://www.facebook.com/wickedelias/posts/2967000120196980
Fair dealing for purpose of parody or satire

© 2022 Red Hat, Inc.

Except where otherwise noted this work is licensed under

http://creativecommons.org/licenses/by/4.0/

Slides	speakerdeck.com/frasertweedale
Blog	frasertweedale.github.io/blog-redhat
Email	`ftweedal@redhat.com`
Twitter	@hackuador