Container Security and new container technologies

Dan Walsh @rhatdan

Distinguished Engineer - Red Hat

Please Stand

Please read out loud all text in RED

I Promise

To say
Make a copy
Rather than
Make a Xerox

I Promise

To say

Tissue

Rather than

Kleenex

I Promise

To say

Container Registries

Rather than

Docker registries

I Promise

To say

Container Images

Rather than

Docker images

I Promise

To say

Containers

Rather than

Docker Containers

Sit Down

What do you need to run a container

- Standard Definition of what makes up a container image.
    - OCI Image Bundle Definition

CoreOS

Introducing Skopeo

https://github.com/containers/skopeo

#nobigfatdaemons

Skopeo

-
  ```
  $ skopeo inspect docker://docker.io/fedora
  ```

-
  ```
  $ skopeo copy docker://busybox:1-glibc atomic:myns/unsigned:streaming
  $ skopeo copy docker://busybox:latest dir:existingemptydirectory
  $ skopeo copy docker://busybox:latest oci:busybox_ocilayout:latest
  ```

-
  ```
  $ skopeo delete docker://localhost:5000/imagename:latest
  ```

What do you need to run a container`

- Standard Definition of what makes up a container image.
    - OCI Image Bundle Definition
- Mechanism to pull images from a container registry to the host
    - github.com/containers/image

Core OS

What do you need to run a container

- Standard Definition of what makes up a container image.
    - OCI Image Bundle Definition
- Mechanism to pull images from a container registry to the host
    - github.com/containers/image
- Ability to explode images onto COW file systems on disk
    - github.com/containers/storage

CoreOS

What do you need to run a container

- Standard Definition of what makes up a container image.
  - OCI Image Bundle Definition
- Mechanism to pull images from a container registry to the host
  - github.com/containers/image
- Ability to explode images onto COW file systems on disk
  - github.com/containers/storage
- Standard mechanism for running a container
  - OCI Runtime Spec (1.0)
  - runc default implementation of OCI Runtime Spec (Same tool Docker uses to run containers)

#nobigfatdaemons

OPENSHIFT

OPENSHIFT

#nobigfatdaemons

What does OpenShift/Kubernetes need run a container?

CRI - Container Runtime Interface

CoreOS

#nobigfatdaemons

What does OpenShift/Kubernetes need run a container?

CRI - Container Runtime Interface

CoreOS

Kubernetes tells CRI to run Container Image:

#nobigfatdaemons

What does OpenShift/Kubernetes need run a container?

CRI - Container Runtime Interface

Kubernetes tells CRI to run Container Image:

- CRI needs to pull image from Container Registry

#nobigfatdaemons

What does OpenShift/Kubernetes need run a container?

CRI - Container Runtime Interface

CoreOS

Kubernetes tells CRI to run Container Image:

- CRI needs to pull image from Container Registry
- CRI Needs to store image on COW File system

What does OpenShift/Kubernetes need run a container?

CRI - Container Runtime Interface

Kubernetes tells CRI to run Container Image:

- CRI needs to pull image from Container Registry
- CRI Needs to store image on COW File system
- CRI Needs to execute OCI Runtime

#nobigfatdaemons

Introducing CRI-O

#nobigfatdaemons

Introducing CRI-O

CRI-O - OCI-based implementation of Kubernetes Container Runtime Interface

- Scope tied to kubernetes CRI
- Only supported user is kubernetes
- Uses standard components as building blocks

“Nothing more, Nothing Less”

#nobigfatdaemons

cri-o

❤

kubernetes

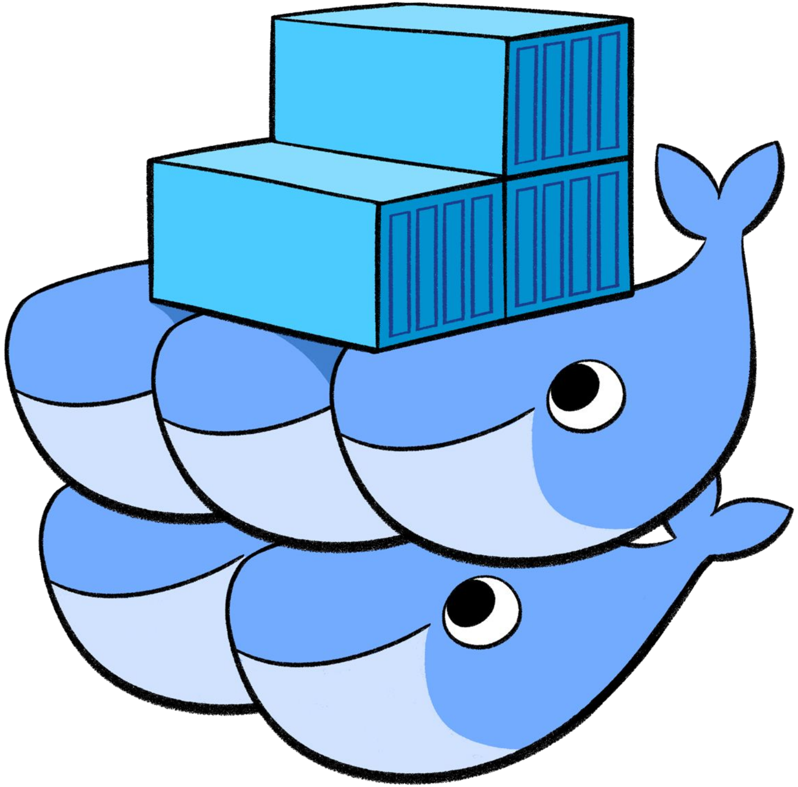
#nobigfatdaemons

MESOSPHERE

#nobigfatdaemons

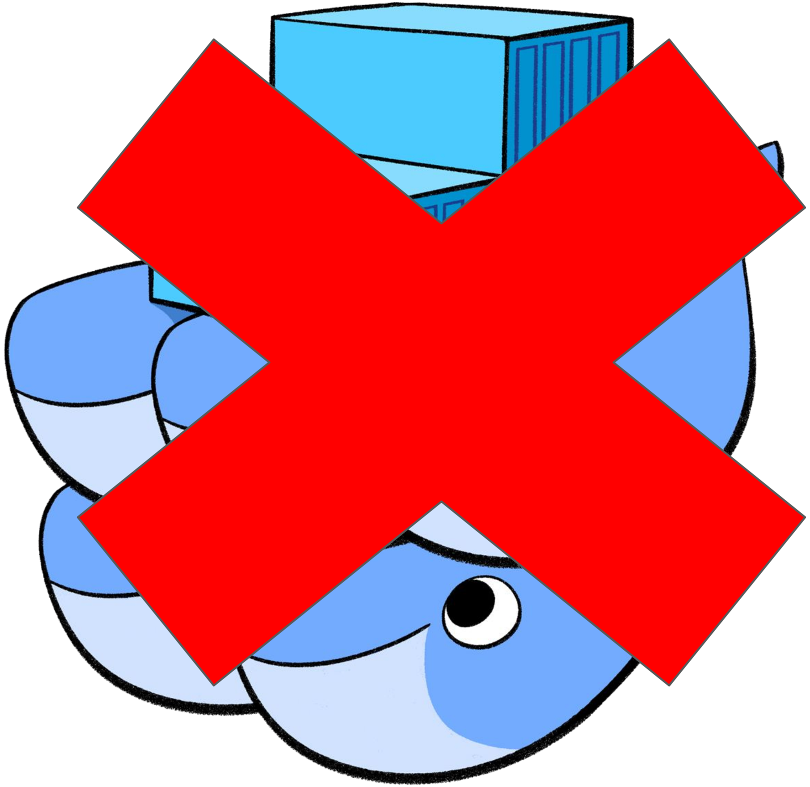
#nobigfatdaemons

SWARM



#nobigfatdaemons

S
W
A
R
M



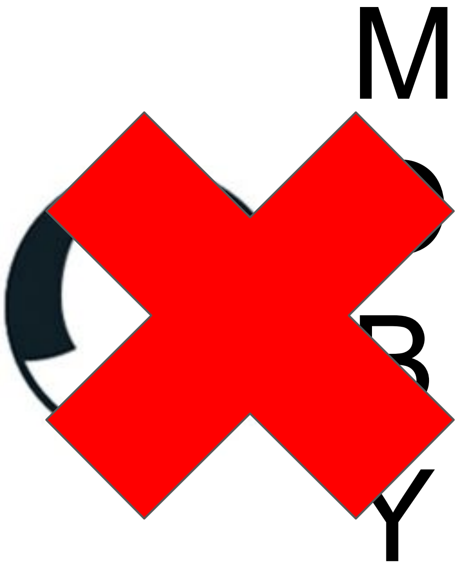
#nobigfatdaemons

vs.

MOBY

#nobigfatdaemons

vs.



#nobigfatdaemons

#nobigfatdaemons

cri-o

❤

kubernetes

#nobigfatdaemons

Overview of additional components

- **oci-runtime-tools** library is used to generate OCI configs for containers

#nobigfatdaemons

Overview of additional components

- **oci-runtime-tools** library is used to generate OCI configs for containers
- **CNI** is used for setting up networking
  - Tested with Flannel, Weave and openshift-sdn

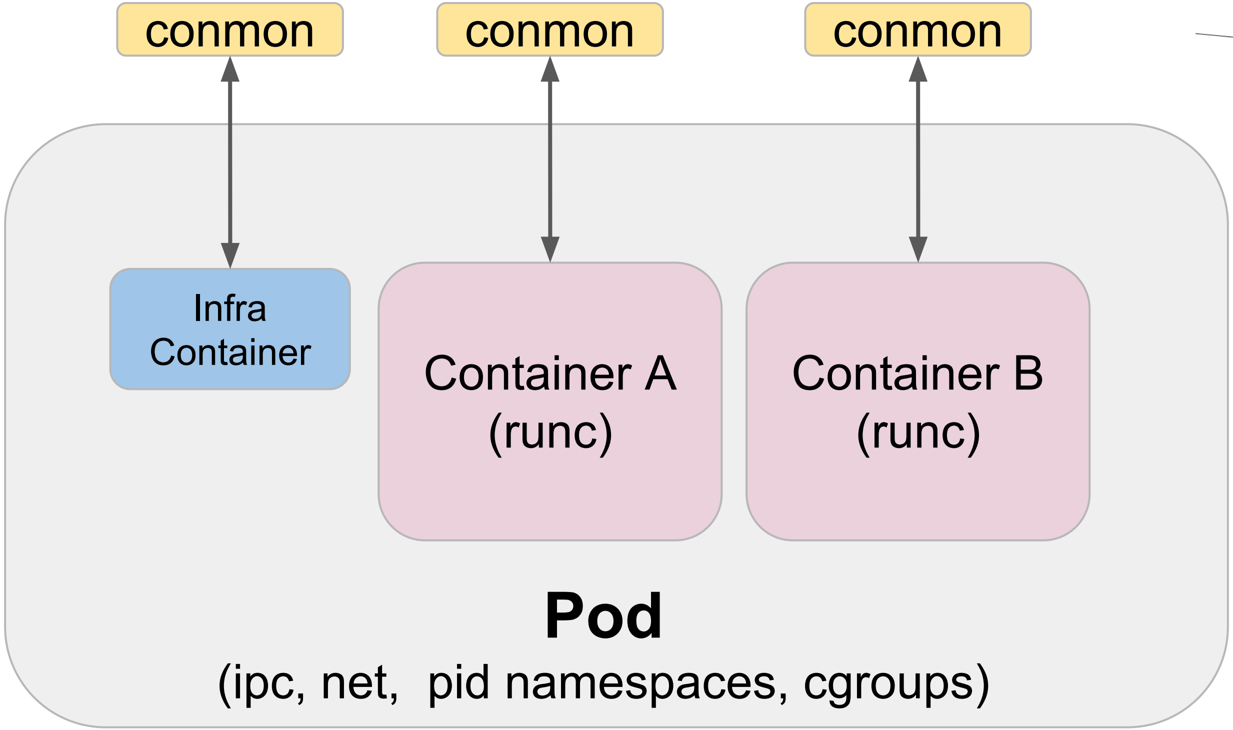
CoreOS

Overview of additional components

- **oci-runtime-tools** library is used to generate OCI configs for containers
- **CNI** is used for setting up networking
  - Tested with Flannel, Weave and openshift-sdn
- **conmon** is a utility for:
  - Monitoring
  - Logging
  - Handling tty
  - Serving attach clients
  - Detecting and reporting OOM

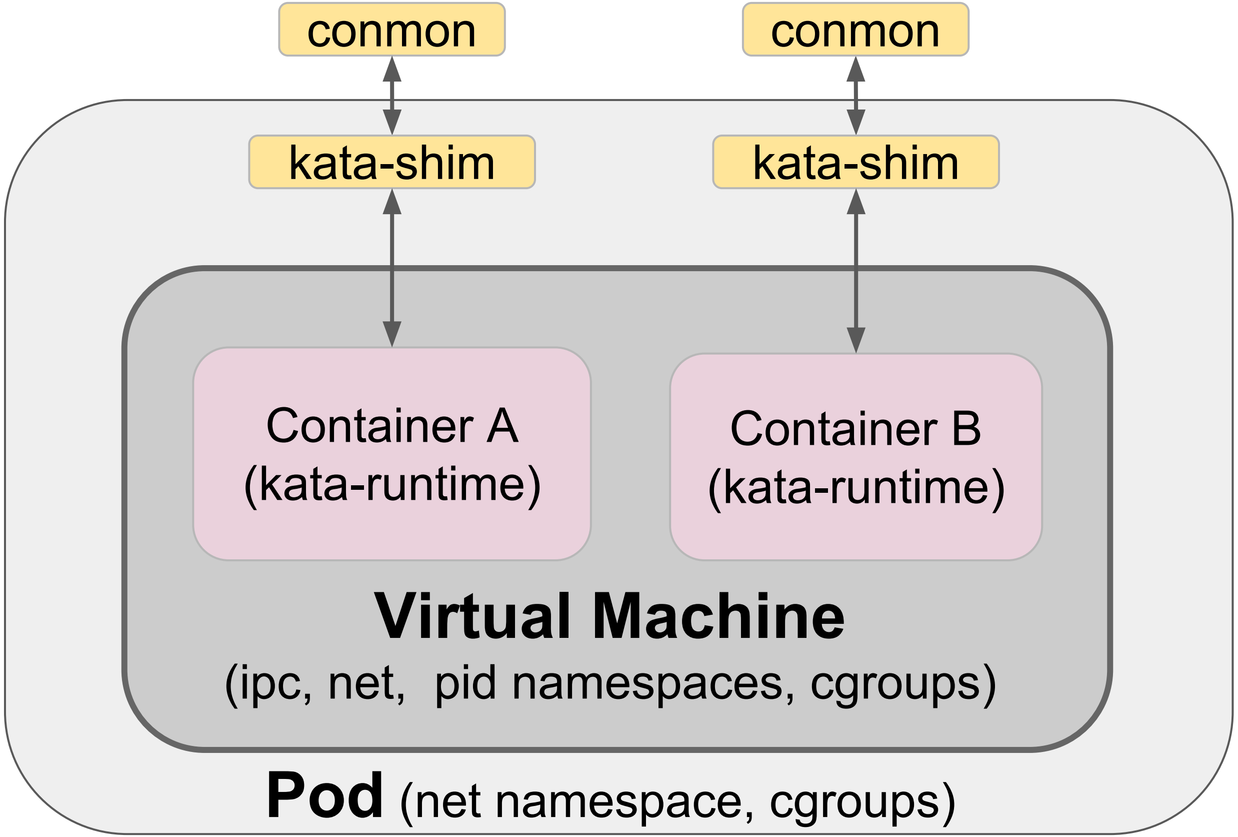
#nobigfatdaemons

Pod architecture (runc)



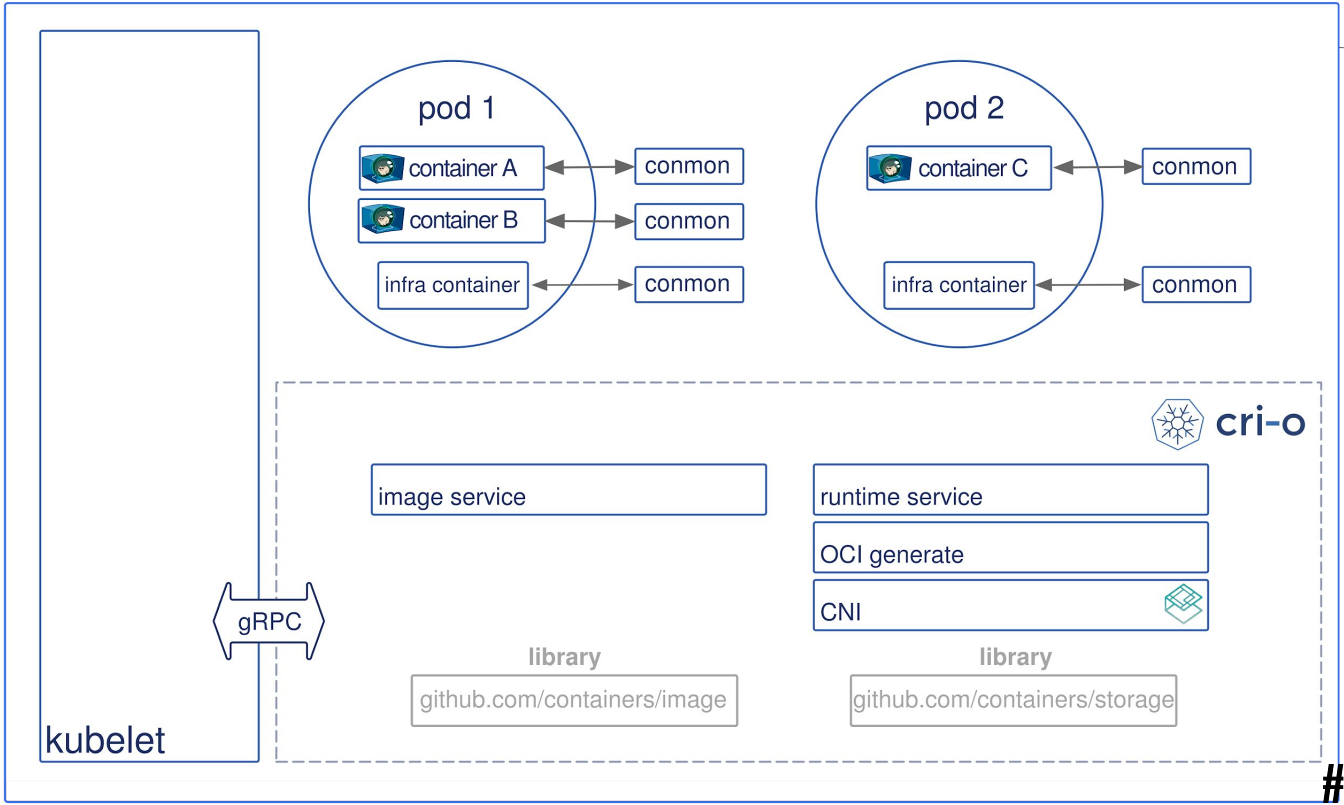
#nobigfatdaemons

Pod architecture (Kata Containers)



#nobigfatdaemons

Architecture



#nobigfatdaemons

Status

- **All** e2e, cri-tools, integration, 9 test suites, (>500) tests passing.
  - **No PRs merged without passing all the tests**

#nobigfatdaemons

Status

- **All** e2e, cri-tools, integration, 9 test suites, (>500) tests passing.
    - **No PRs merged without passing all the tests.**
- 1.0.7 (kube 1.7.x) supported. (December 2017)

#nobigfatdaemons

Status

- **All** e2e, cri-tools, integration, 9 test suites, (>500) tests passing.
  - **No PRs merged without passing all the tests.**
- 1.0.7 (kube 1.7.x) supported. (December 2017)
- 1.9.12 (kube 1.9.x) released.
  - CRI-O fully supported in OpenShift 3.9 along with docker.

#nobigfatdaemons

Status

- **All** e2e, cri-tools, integration, 9 test suites, (>500) tests passing.
  - **No PRs merged without passing all the tests.**
- 1.0.7 (kube 1.7.x) supported. (December 2017)
- 1.9.12 (kube 1.9.x) released.
  - CRI-O fully supported in OpenShift 3.9 along with docker.
- 1.10.6 (kube 1.10.x) released.

#nobigfatdaemons

Status

- **All** e2e, cri-tools, integration, 9 test suites, (>500) tests passing.
  - **No PRs merged without passing all the tests.**
- 1.0.7 (kube 1.7.x) supported. (December 2017)
- 1.9.12 (kube 1.9.x) released.
  - CRI-O fully supported in OpenShift 3.9 along with docker.
- 1.10.6 (kube 1.10.x) released.
- 1.11.2 (Kube 1.11.x) released

#nobigfatdaemons

Status

- **All** e2e, cri-tools, integration, 9 test suites, (>500) tests passing.
  - **No PRs merged without passing all the tests.**
- 1.0.7 (kube 1.7.x) supported. (December 2017)
- 1.9.12 (kube 1.9.x) released.
  - CRI-O fully supported in OpenShift 3.9 along with docker.
- 1.10.6 (kube 1.10.x) released.
- 1.11.2 (Kube 1.11.x) released
- 1.12.1 (Kube 1.12.x) released
- Goal for Openshift 4.0 is to fully support CRI-O by default.

#nobigfatdaemons

Status

CRI-O is now powering nodes on OpenShift Online.

#nobigfatdaemons

" CRI-O just works for them,

so they haven’t had much to say"

#nobigfatdaemons

Making running containers in production

boring

#nobigfatdaemons

Security in CRI-O

- No Hard-Coded Capabilities list
    - Since CRI-O does not do builds, containers by default have less capabilities

#nobigfatdaemons

Security in CRI-O

- No Hard-Coded Capabilities list
  - Since CRI-O does not do builds, containers by default have less capabilities
- Read Only Containers
  - In production containers should not be allowed to modify images

#nobigfatdaemons

Security in CRI-O

- No Hard-Coded Capabilities list
  - Since CRI-O does not do builds, containers by default have less capabilities
- Read Only Containers
  - In production containers should not be allowed to modify images
- Kata Containers support

#nobigfatdaemons

Security in CRI-O

- No Hard-Coded Capabilities list
  - Since CRI-O does not do builds, containers by default have less capabilities
- Read Only Containers
  - In production containers should not be allowed to modify images
- Kata Containers support
- Better User Namespace support

#nobigfatdaemons

What else does OpenShift need?

- Ability to build container images
- Ability to push container images to container registries

OPENSHIFT

#nobigfatdaemons

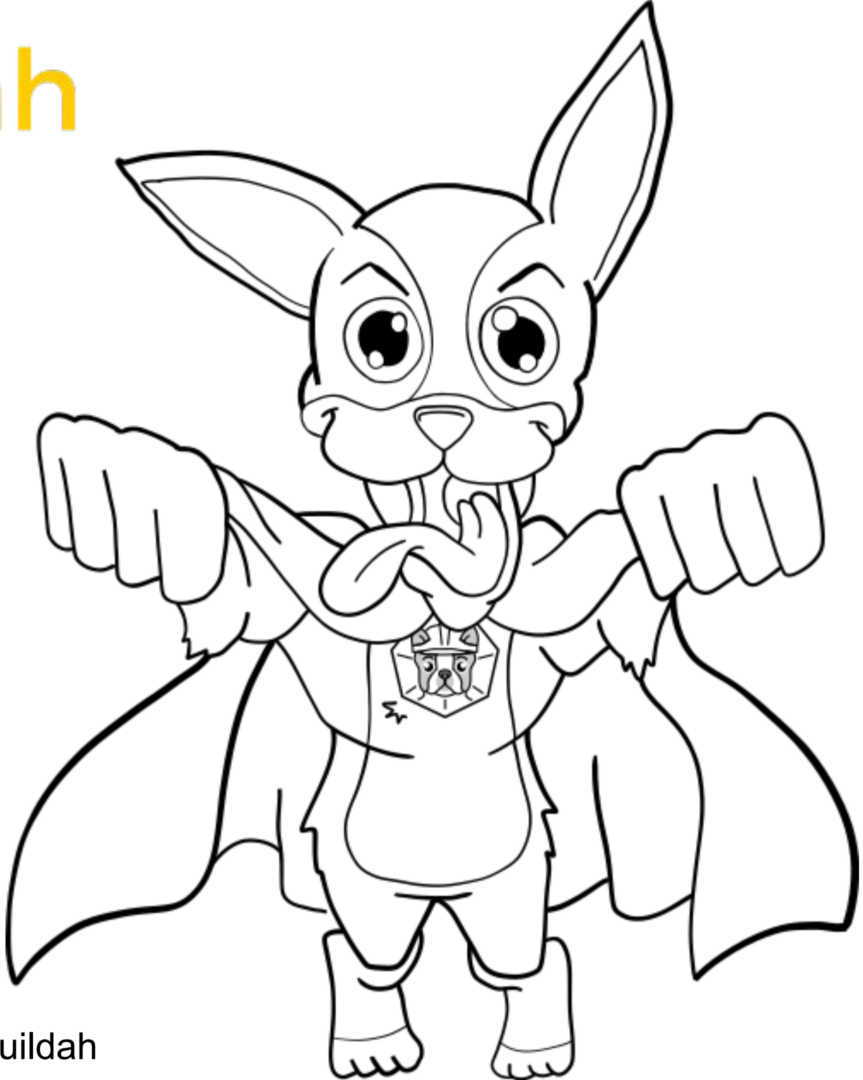
#nobigfatdaemons

Introducing Buildah

https://github.com/containers/buildah

#nobigfatdaemons

buildah



https://github.com/containers/buildah

#nobigfatdaemons

buildah

#nobigfatdaemons

buildah

Coreutils for building containers. Simple interface

#nobigfatdaemons

buildah

Coreutils for building containers. Simple interface

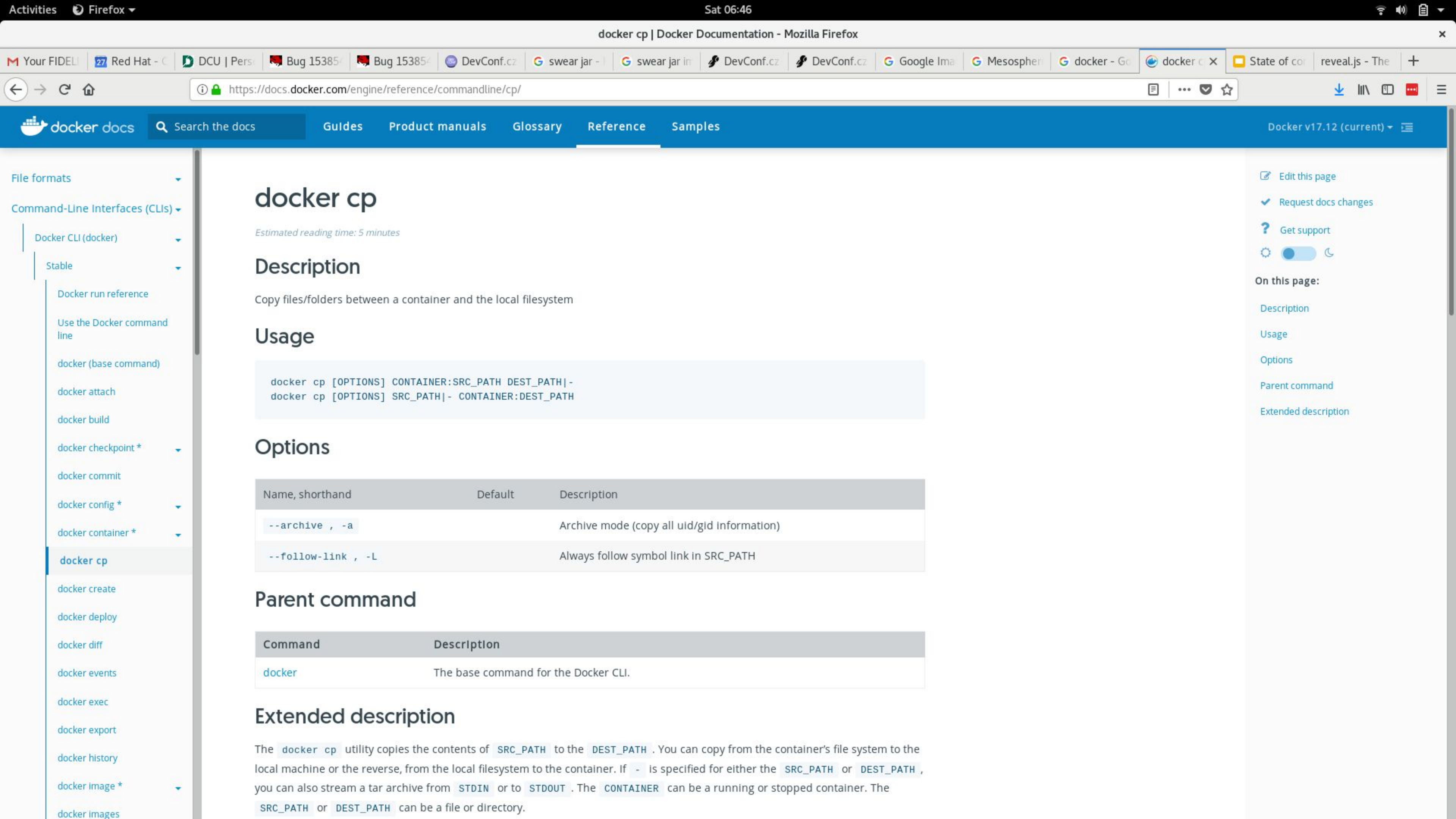
```
# ctr=$(buildah from fedora)
```

buildah

Coreutils for building containers. Simple interface

```
# ctr=$(buildah from fedora)
# mnt=$(buildah mount $ctr)
```

#nobigfatdaemons



docker cp

Estimated reading time: 5 minutes

Description

Copy files/folders between a container and the local filesystem

Usage

```
docker cp [OPTIONS] CONTAINER:SRC_PATH DEST_PATH|-
docker cp [OPTIONS] SRC_PATH|- CONTAINER:DEST_PATH
```

Options

Name, shorthand	Default	Description
`--archive , -a`		Archive mode (copy all uid/gid information)
`--follow-link , -L`		Always follow symbol link in SRC_PATH

Parent command

Command	Description
docker	The base command for the Docker CLI.

Extended description

The `docker cp` utility copies the contents of `SRC_PATH` to the `DEST_PATH`. You can copy from the container's file system to the local machine or the reverse, from the local filesystem to the container. If `-` is specified for either the `SRC_PATH` or `DEST_PATH`, you can also stream a tar archive from `STDIN` or to `STDOUT`. The `CONTAINER` can be a running or stopped container. The `SRC_PATH` or `DEST_PATH` can be a file or directory.

buildah

Coreutils for building containers. Simple interface

```
# ctr=$(buildah from fedora)
# mnt=$(buildah mount $ctr)
# cp -R src $mnt
```

#nobigfatdaemons

buildah

Coreutils for building containers. Simple interface

```
# ctr=$(buildah from fedora)
# mnt=$(buildah mount $ctr)
# cp -R src $mnt
# dnf install --installroot=$mnt httpd
```

#nobigfatdaemons

buildah

Coreutils for building containers. Simple interface

```
# ctr=$(buildah from fedora)
# mnt=$(buildah mount $ctr)
# cp -R src $mnt
# dnf install --installroot=$mnt httpd
# make install DESTDIR=$mnt
```

#nobigfatdaemons

buildah

Coreutils for building containers. Simple interface

```
# ctr=$(buildah from fedora)
# mnt=$(buildah mount $ctr)
# cp -R src $mnt
# dnf install --installroot=$mnt httpd
# make install DESTDIR=$mnt
# buildah config --entrypoint=/usr/sbin/test.sh --env foo=bar $ctr
```

#nobigfatdaemons

buildah

Coreutils for building containers. Simple interface

```
# ctr=$(buildah from fedora)
# mnt=$(buildah mount $ctr)
# cp -R src $mnt
# dnf install --installroot=$mnt httpd
# make install DESTDIR=$mnt
# buildah config --entrypoint=/usr/sbin/test.sh --env foo=bar $ctr
# buildah commit $ctr myhttpd
```

#nobigfatdaemons

buildah

Coreutils for building containers. Simple interface

```
# ctr=$(buildah from fedora)
# mnt=$(buildah mount $ctr)
# cp -R src $mnt
# dnf install --installroot=$mnt httpd
# make install DESTDIR=$mnt
# buildah config --entrypoint=/usr/sbin/test.sh --env foo=bar $ctr
# buildah commit $ctr myhttpd
# buildah push myhttpd docker://rhatdan/myhttpd
```

#nobigfatdaemons

buildah

Dan Wait!

buildah

Dan Wait!
What about Dockerfile?????

#nobigfatdaemons

buildah

Buildah also supports Dockerfile

```
buildah build-using-dockerfile -f Dockerfile .
```

#nobigfatdaemons

buildah

Buildah also supports Dockerfile
buildah build-using-dockerfile -f Dockerfile .
Or for those lazy ones:
buildah **bud** -f Dockerfile .

#nobigfatdaemons

buildah

Does Buildah have a scripting language? Perhaps Buildahfile?

#nobigfatdaemons

buildah

BASH

BASH

We want others to build higher level tools on Buildah.

#nobigfatdaemons

BASH

We want others to build higher level tools on Buildah.
Working to make OpenShift use Buildah for S2I containers rather then use Docker.

buildah

BASH

We want others to build higher level tools on Buildah.
Working to make OpenShift use Buildah for S2I containers rather then use Docker.
Want to work with Ansible-containers to use buildah for containers as well.

#nobigfatdaemons

buildah

Security

- No Big Fat Container Daemon
  - Run your container builds inside of locked down containers under Kubernetes
  - No need to leak in the docker.sock

#nobigfatdaemons

buildah

Security

- No Big Fat Container Daemon
  - Run your container builds inside of locked down containers under Kubernetes
  - No need to leak in the docker.sock
- Working on running as non root from desktop

#nobigfatdaemons

buildah

Security

- No Big Fat Container Daemon
    - Run your container builds inside of locked down containers under Kubernetes
    - No need to leak in the docker.sock
- Working on running as non root from desktop
- Building Minimal Images
    - Only include content in the image required to run the image
    - Does not require you to use Dockerfile and therefore include Yum/Python in image

#nobigfatdaemons

What else does OpenShift need?

- Ability to diagnose problems on the host
- If you don’t use Docker to run the containers, how does an admin discover what is going on in his Container runtime, without the docker CLI?

#nobigfatdaemons

Introducing podman
part of the libpod effort

#nobigfatdaemons

Replacing Docker With Podman

By Dan Walsh @rhatdan

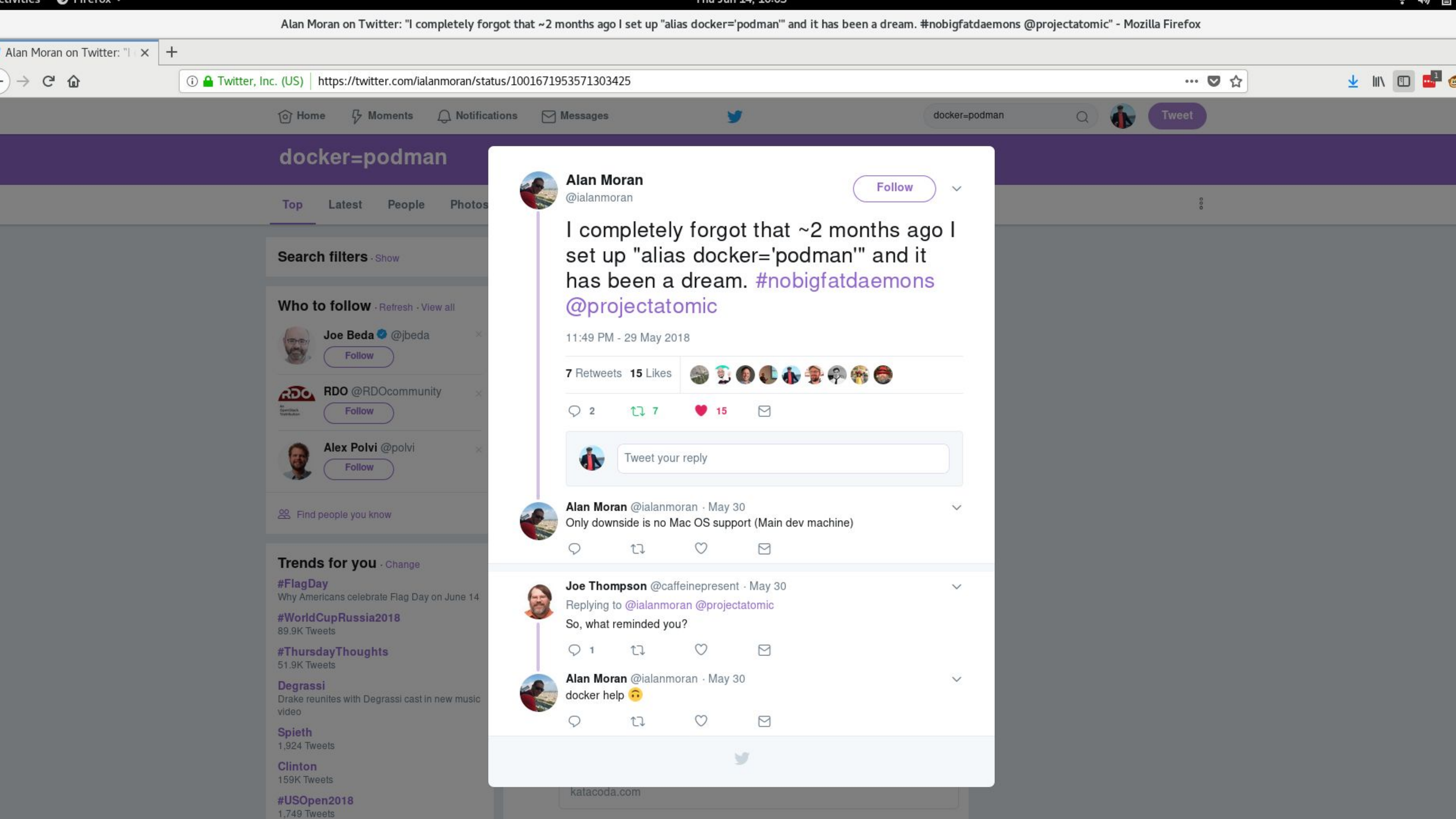
```
dnf install -y podman
dnf install -y podman

alias docker=podman
```

Questions

Blog: https://medium.com/cri-o
Github:

- https://github.com/kubernetes-sigs/cri-o
- https://github.com/containers/buildah
- https://github.com/containers/skopeo
- https://github.com/containers/libpod (podman)
- https://github.com/containers/storage
- https://github.com/containers/image

Site: https://cri-o.io IRC: freenode: #cri-o
Site: https://podman.io IRC: freenode: #podman
Site: https://buildah.io IRC: freenode: #buildah



Alan Moran @ialanmoran

I completely forgot that ~2 months ago I set up "alias docker='podman'" and it has been a dream. #nobigfatdaemons @projectatomic

11:49 PM - 29 May 2018

7 Retweets 15 Likes

2 7 15

Alan Moran @ialanmoran · May 30

Only downside is no Mac OS support (Main dev machine)

Joe Thompson @caffeinepresent · May 30

Replying to @ialanmoran @projectatomic

So, what reminded you?

1

Alan Moran @ialanmoran · May 30

docker help 🙃

Introducing podman

podman is tool for managing POD/Containers based on the Docker CLI

https://github.com/projectatomic/libpod

#nobigfatdaemons

Introducing podman

podman is tool for managing POD/Containers based on the Docker CLI

```
# podman ps -a
```

https://github.com/projectatomic/libpod

#nobigfatdaemons

Introducing podman

podman is tool for managing POD/Containers based on the Docker CLI

```
# podman ps -a
```

```
# podman run -ti fedora sleep 2000
```

https://github.com/projectatomic/libpod

#nobigfatdaemons

Introducing podman

podman is tool for managing POD/Containers based on the Docker CLI

```
# podman ps -a

# podman run -ti fedora sleep 2000

# podman exec -ti fedora sh
```

https://github.com/projectatomic/libpod

#nobigfatdaemons

Introducing podman

podman is tool for managing POD/Containers based on the Docker CLI

```
# podman ps -a

# podman run -ti fedora sleep 2000

# podman exec -ti fedora sh

# podman images
```

...

https://github.com/projectatomic/libpod

#nobigfatdaemons

DEMO

#nobigfatdaemons

Security

- No Big Fat Container Daemon
  - No need to leak in the docker.sock
  - Run Manage/Containers without being root.
  - No need for access to the /var/run/docker.sock

#nobigfatdaemons

Security

- No Big Fat Container Daemon
    - No need to leak in the docker.sock
    - Run Manage/Containers without being root.
    - No need for access to the /var/run/docker.sock
- Containers run as child of the process that ran it
    - Better Auditing
    - Support for socket activation

#nobigfatdaemons

Proper Integration with Systemd

- Can run systemd as PID 1 in container, with no modifications

Proper Integration with Systemd

- Can run systemd as PID 1 in container, with no modifications
- Support sd_notify

Proper Integration with Systemd

- Can run systemd as PID 1 in container, with no modifications
- Support sd_notify
- Socket Activation

Remote API for Podman

- Added Varlink support
- Socket activation of podman system service with varlink

```
[Unit]
Description=Podman Remote API Service
Requires=io.podman.socket
After=io.podman.socket
Documentation=man:podman-varlink(1)

[Service]
Type=simple
ExecStart=/usr/bin/podman varlink unix:/run/podman/io.podman

[Install]
WantedBy=multi-user.target
Also=io.podman.socket
```

Python Bindings

```
python3 -c "import podman; import json; c=podman.Client();print(json.dumps(c.system.info(), indent=4))"
[
    {
        "mem_free": 5796605952,
        "mem_total": 16679206912,
        "swap_free": 0,
        "swap_total": 0,
        "arch": "amd64",
        "cpus": 4,
        "hostname": "localhost.localdomain",
        "kernel": "4.18.9-200.fc28.x86_64",
        "os": "linux",
        "uptime": "11h 2m 32.25s (Approximately 0.46 days)"
    },
...
```

Remote API Support

pypodman - Python program used for running remote podman commands.

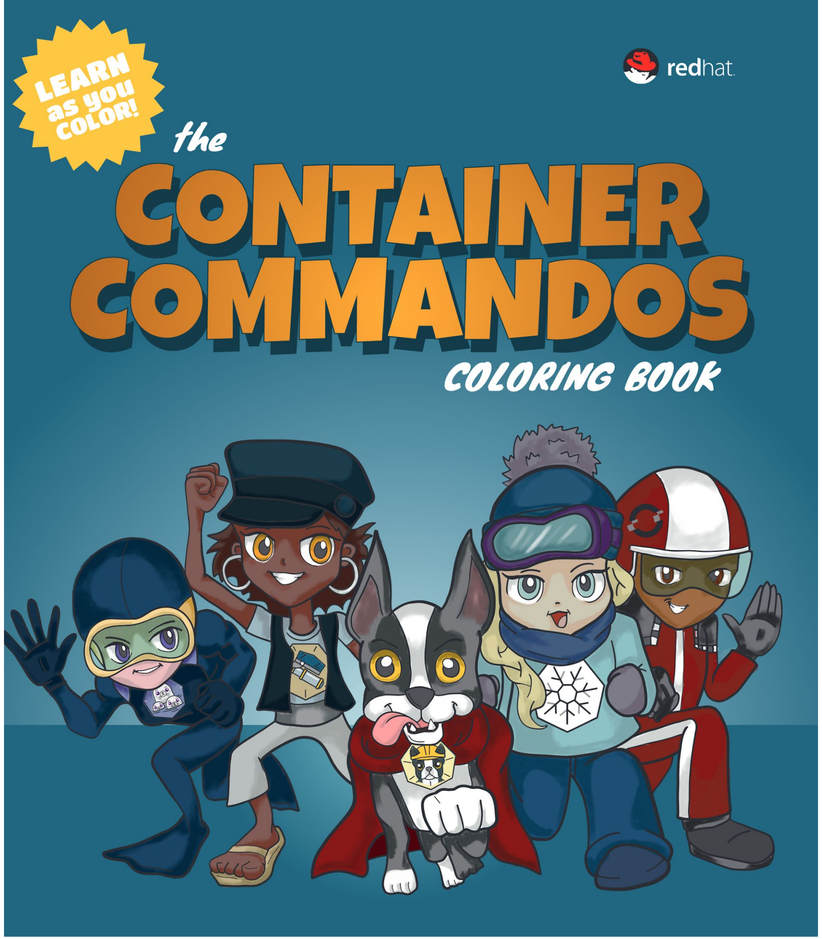
https://asciinema.org/a/203590

Cockpit support

https://github.com/cockpit-project/cockpit-podman

What we don’t do

- Autostart, autorestart
  - Systemd should be handling this
- Swarm
  - We support Kubernetes container orchestrator
- Notary
  - We do support simple signing, but would look at PRs for Notary support
- HealthChecks
  - We are looking into this, perhaps systemd support? Side car container in pod?
- Docker API - We have no plans to support this, but we do have Varlink
- Docker volumes
  - It is on the roadmap



https://github.com/mairin/coloringbook-container-commandos/blob/master/Web.pdf

Questions

Blog: https://medium.com/cri-o
Github:

- https://github.com/kubernetes-sigs/cri-o
- https://github.com/containers/buildah
- https://github.com/containers/skopeo
- https://github.com/containers/libpod (podman)
- https://github.com/containers/storage
- https://github.com/containers/image

Site: https://cri-o.io IRC: freenode: #cri-o
Site: https://podman.io IRC: freenode: #podman
Site: https://buildah.io IRC: freenode: #buildah