



Container Security and new container technologies

Dan Walsh @rhatdan

Senior 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

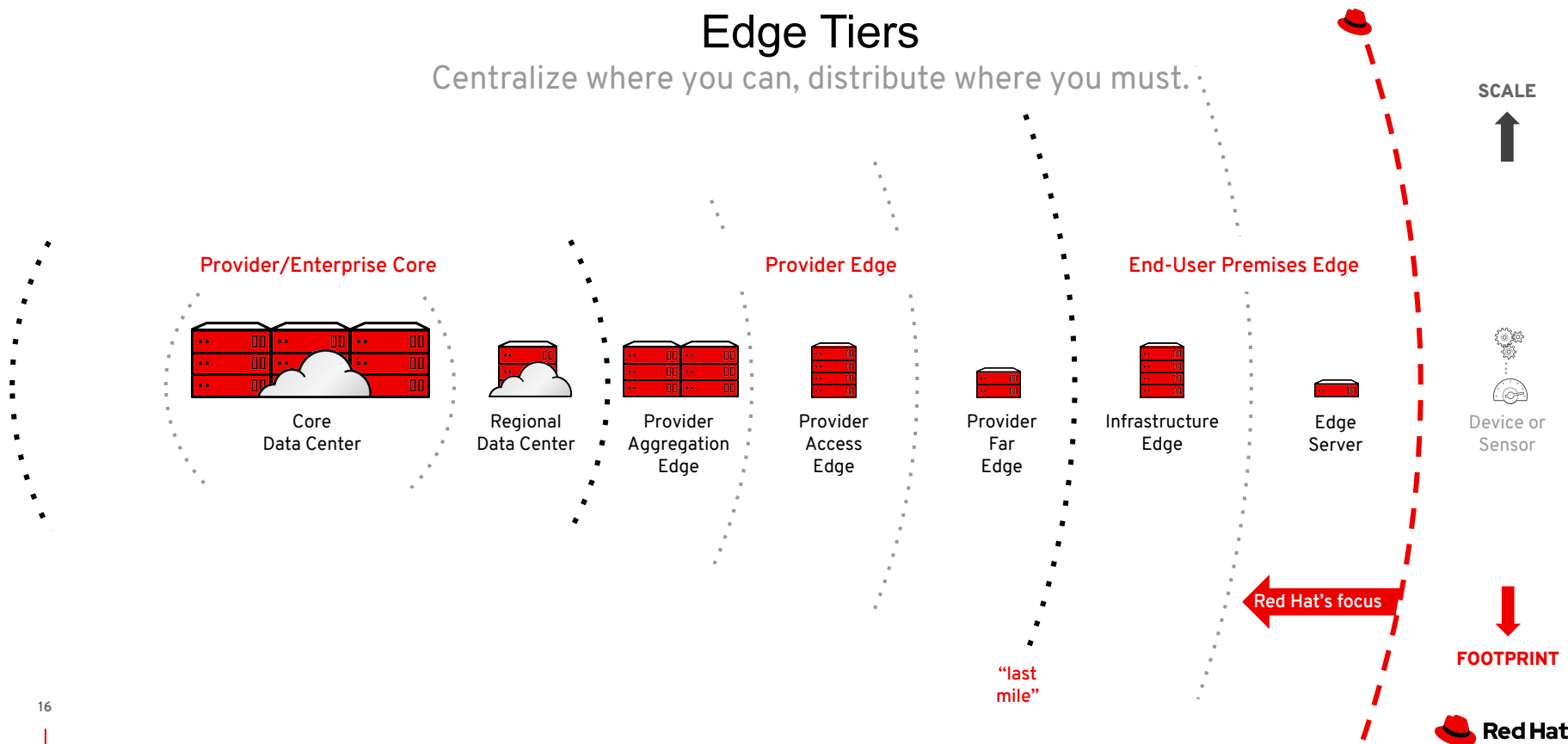
Sit Down



Dan Quiote

Edge Tiers

Centralize where you can, distribute where you must.



Importance of the OS to the Edge



Core OS

What do you need to run a container

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



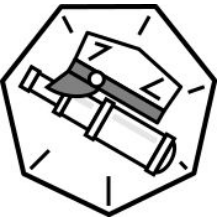
Importance of the Application to the Edge



OPEN

CONTAINER
INITIATIVE



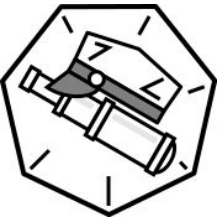


Introducing Skopeo



<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`



#nobigfatdaemons

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



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



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



OPENSIFT

OPENSIFT

#nobigfatdaemons

What does OpenShift/Kubernetes need run a container?

CRI - Container Runtime Interface



#nobigfatdaemons

What does OpenShift/Kubernetes need run a container?

CRI - Container Runtime Interface



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



Kubernetes tells CRI to run Container Image:

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

#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
- CRI Needs to store image on COW File system
- CRI Needs to execute OCI Runtime

#nobigfatdaemons

Introducing CRI-0



#nobigfatdaemons

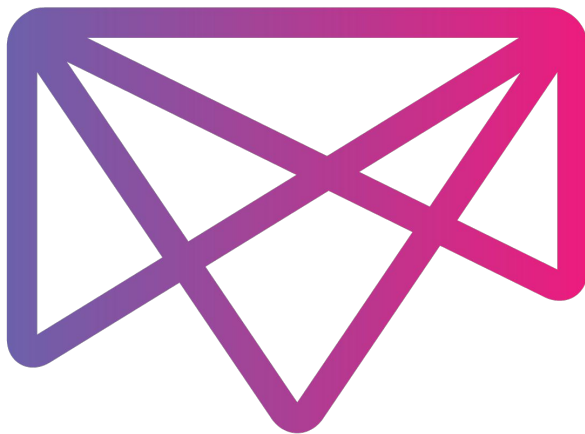


cri-o



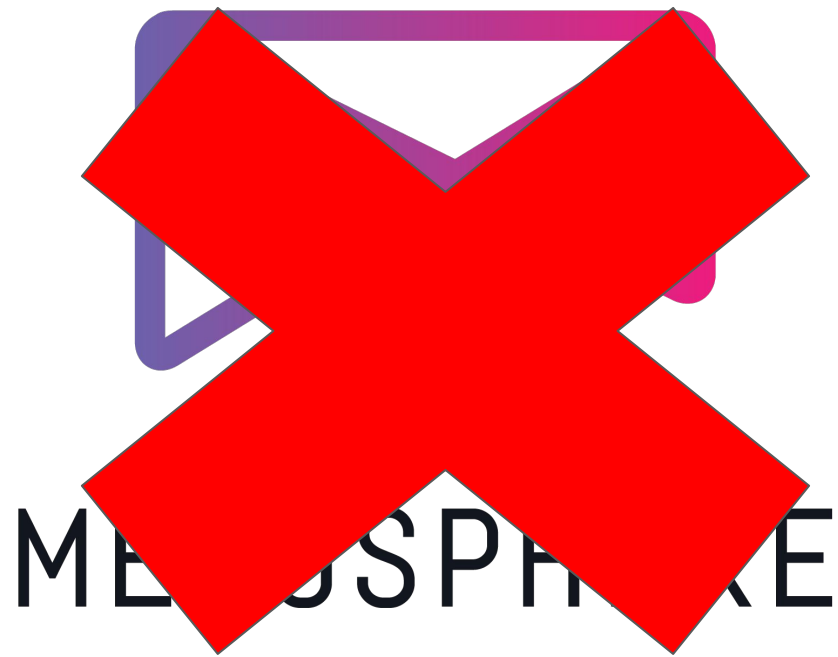
kubernetes

#nobigfatdaemons



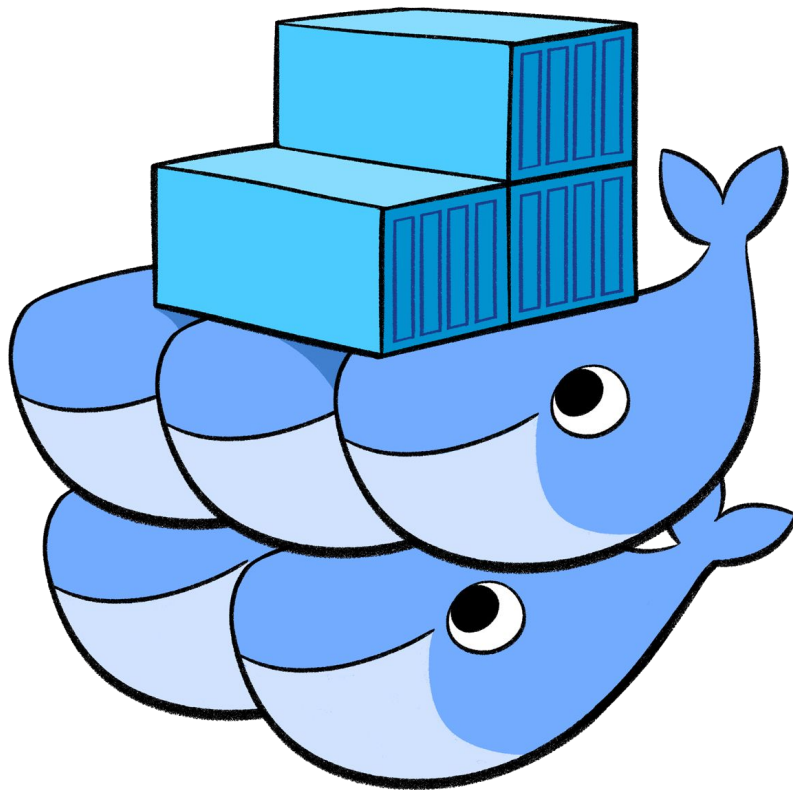
MESOSPHERE

#nbigfatdaemons



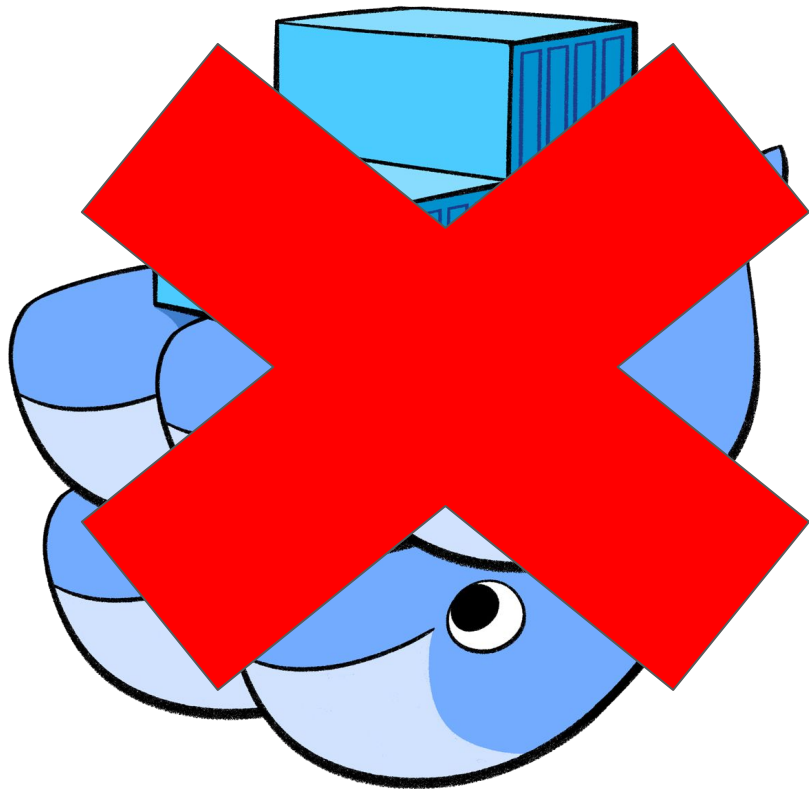
#nobigfatdaemons

S
W
A
R
M



#nobigfatdaemons

S
W
A
R
M



#nobigfatdaemons



vs.

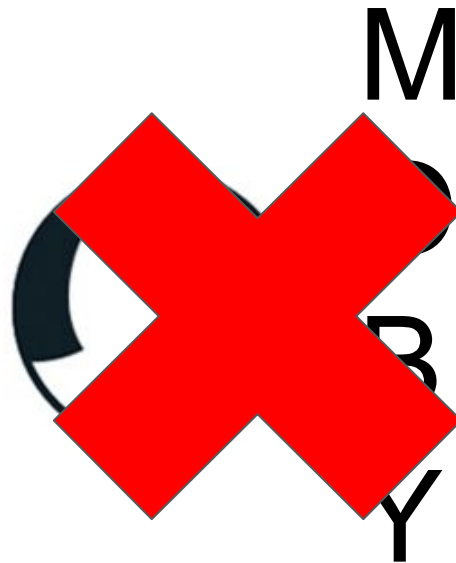


M
O
B
Y

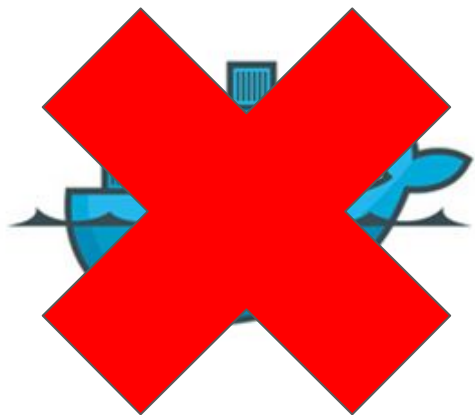
#nobigfatdaemons



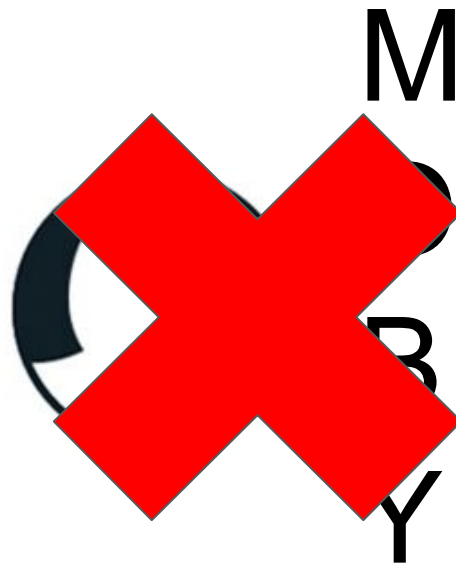
vs.



#nobigfatdaemons



VS.



#nobigfatdaemons



cri-o



kubernetes

#nobigfatdaemons

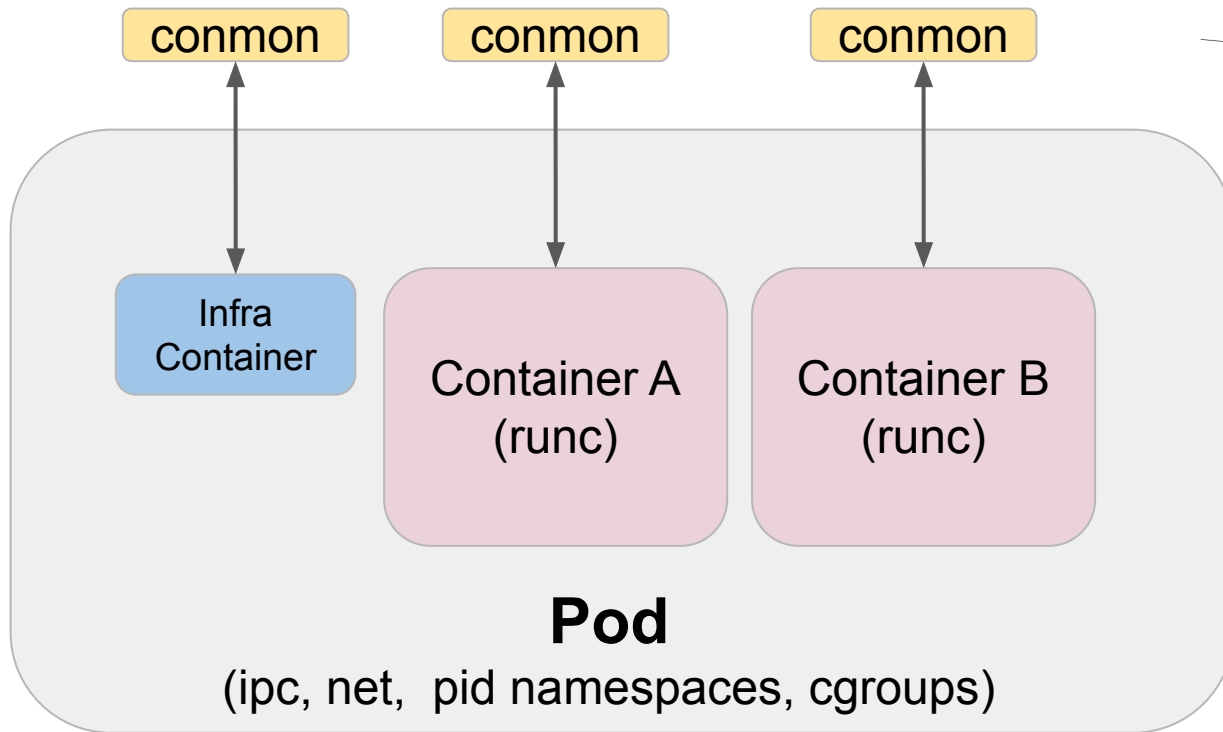
Overview of additional components

- **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
- **common** is a utility for:
 - Monitoring
 - Logging
 - Handling tty
 - Serving attach clients
 - Detecting and reporting OOM



#nobigfatdaemons

Pod architecture (runc)



#nobigfatdaemons

Status

- **All** e2e, cri-tools, integration, 11 test suites, (>1500) tests passing.
 - **No PRs merged without passing all the tests.**
- 1.0.7 (kube 1.7.x) supported. (December 2017)
- 1.9.14 (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
- 1.12.5 (Kube 1.12.x) released
- 1.13.1 (Kuber 1.13.*) released
- 1.14.1 (Kuber 1.14.*) 1.15.5 (Kuber 1.15.*), 1.16.1 (Kuber 1.16.*) ... released



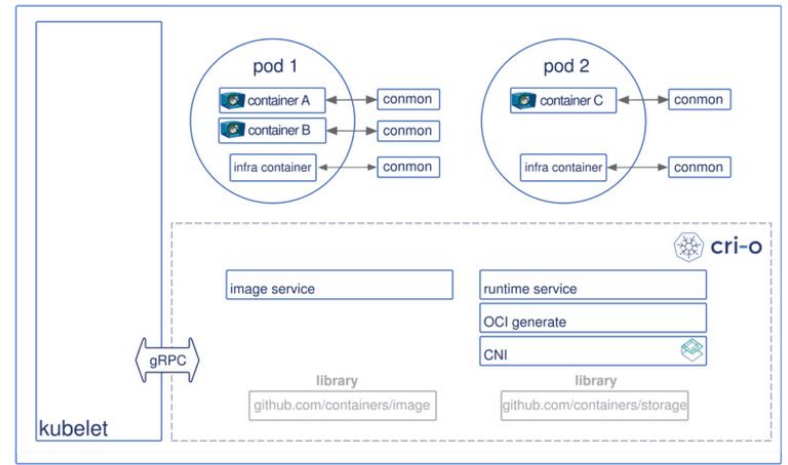
#nobigfatdaemons

CNCF to Host CRI-O

By cncf | Apr 8, 2019 | Blog

0

Today, the Cloud Native Computing Foundation (CNCF) Technical Oversight Committee (TOC) voted to accept CRI-O as an incubation-level hosted project. CRI-O, created by Red Hat, is an implementation of the Kubernetes Container Runtime Interface (CRI) designed to enable the use of Open Container Initiative (OCI) compatible runtime.



"A founding principal of CRI-O was to 'not reinvent the wheel' but to use shared components and refine approaches tested in production, and existing, battle-tested code," said Brendan Burns, TOC representative and project sponsor, Kubernetes cofounder. "As CRI-O is



Openshift 4.0 Uses CRI-O

No Docker

CRI-O is now powering nodes on OpenShift Online

#nobigfatdaemons



Making running containers in production

boring

#nobigfatdaemons

CRI-O Importance for the Edge

- Stability
 - Optimized for Kubernetes workloads.
 - Minimize CPU Utilization
 - Minimize Memory Footprint



#nobigfatdaemons

CRI-O Importance for the Edge

Security

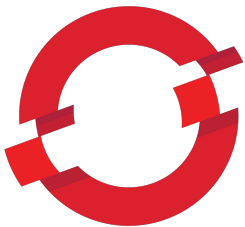
- No Hard-Coded Capabilities list
 - Since CRI-O does not do builds, containers by default have less capabilities
- Read Only Containers
 - In production containers should not be allowed to modify images
- Kata Containers support
- Better User Namespace support



#nobigfatdaemons

What else does OpenShift need?

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



OPENSHIFT

#nobigfatdaemons



#nobigfatdaemons

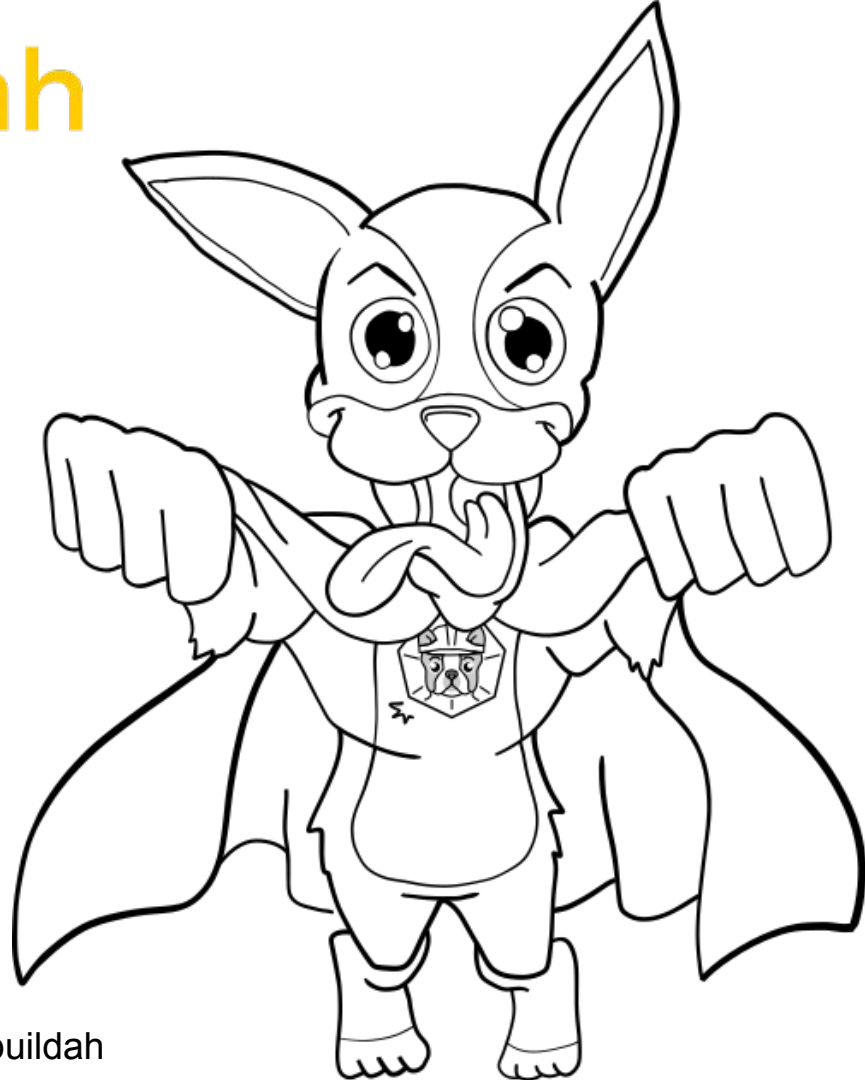
Introducing Buildah



buildah

<https://github.com/containers/buildah>

#nobigfatdaemons



<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)

#nobigfatdaemons



buildah



Coreutils for building containers. Simple interface

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

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

#nobigfatdaemons

Activities

Firefox

Sat 06:46

docker cp | Docker Documentation - Mozilla Firefox

My FIDELI

27 Red Hat - C

DCU | Pers

Bug 15385

Bug 15385

DevConf.cz

swear jar -

swear jar in

DevConf.cz

DevConf.cz

Google Ima

Mesospher

docker - Go

docker (X

State of co

reveal.js - The

+

← → ↺ 🏠

🔒 https://docs.docker.com/engine/reference/commandline/cp/

📄 ⋮ 🍷 ☆

⬇ 📄 📄 📄 📄

⋮

🚢 docker docs

🔍 Search the docs

Guides

Product manuals

Glossary

Reference

Samples

Docker v17.12 (current) ▾ 📄

File formats ▾

Command-Line Interfaces (CLIs) ▾

Docker CLI (docker) ▾

Stable ▾

Docker run reference

Use the Docker command line

docker (base command)

docker attach

docker build

docker checkpoint *

docker commit

docker config *

docker container *

docker cp

docker create

docker deploy

docker diff

docker events

docker exec

docker export

docker history

docker image *

docker images

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
<code>--archive , -a</code>		Archive mode (copy all uid/gid information)
<code>--follow-link , -L</code>		Always follow symbol link in SRC_PATH

Parent command

Command	Description
<code>docker</code>	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.

✎ Edit this page

✓ Request docs changes

? Get support

⚙ ☒ ☐

On this page:

Description

Usage

Options

Parent command

Extended description



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!

#nobigfatdaemons



buildah



Dan Wait!

What about Dockerfile?????

#nbigfatdaemons



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



#nobigfatdaemons



buildah



BASH

We want others to build higher level tools on Buildah.

Working to make OpenShift use Buildah for S2I containers rather than use Docker.

Want to work with Ansible-containers to use buildah for containers as well.

#nobigfatdaemons



buildah



Buildah Importance for the Edge

#nobigfatdaemons



buildah



Buildah Importance for the Edge

None

#nobigfatdaemons



buildah



Buildah Importance for the Edge

Container images - critical component of shipping software on edge.

#nbigfatdaemons



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
- Buildah can be run as non root on the 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



#nbigfatdaemons

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:

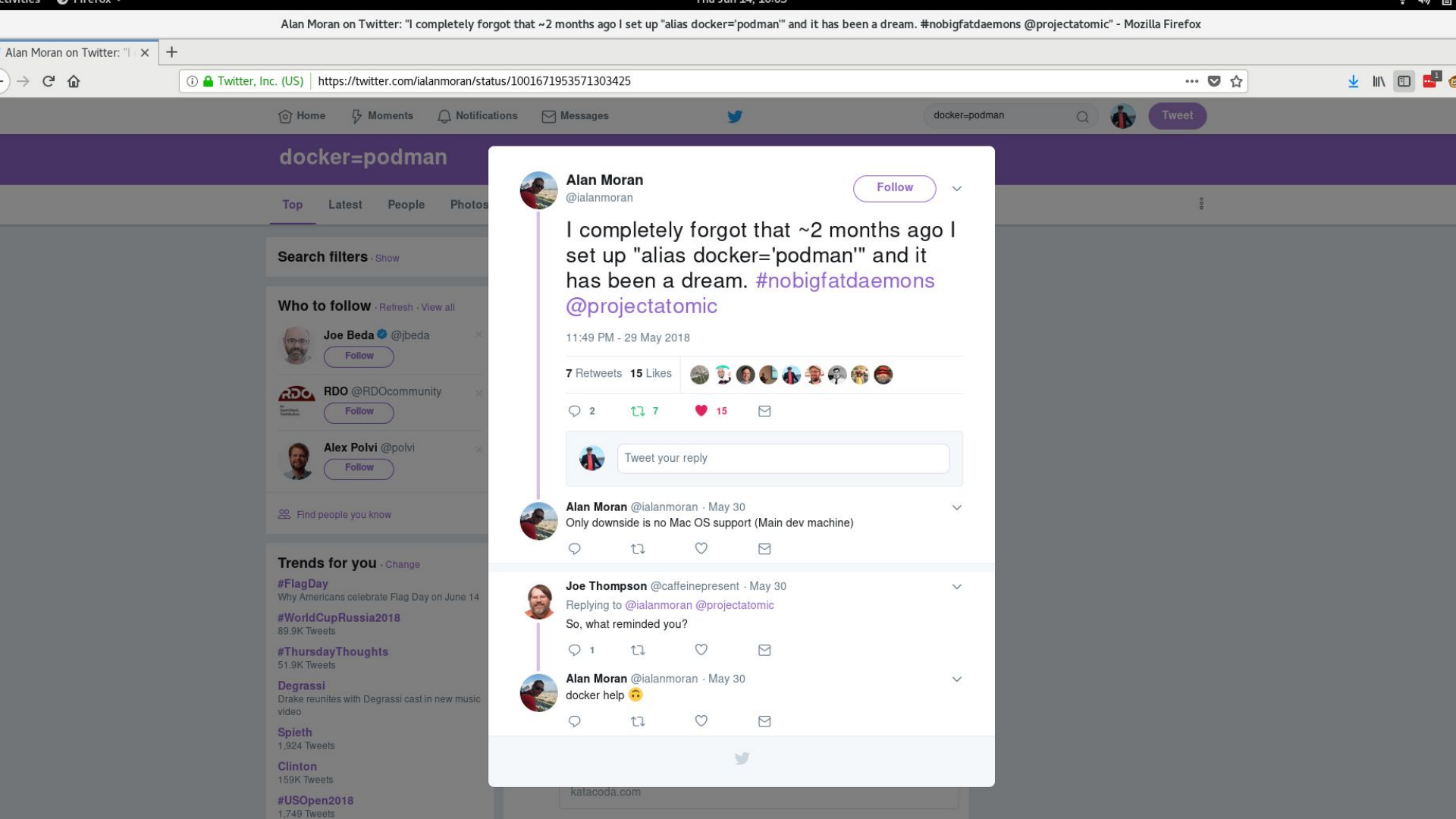
- <https://github.com/kubernetes-sigs/cri-o>
- <https://github.com/containers/buildah>
- <https://github.com/containers/skopeo>
- <https://github.com/containers/libpod> (podman)
- <https://github.com/containers/storage>
- <https://github.com/containers/image>

Site: <https://cri-o.io> IRC: freenode: #cri-o

Site: <https://podman.io> IRC: freenode: #podman

Site: <https://buildah.io> IRC: freenode: #buildah







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



Podman on the edge



Supportability

- No Big Fat Container Daemon
 - Why run a container daemon when you only want to run a single application
- Integration with Systemd
 - Able to allow systemd and operating system manage lifecycle of containerized application

#nobigfatdaemons



Podman on the edge

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



Red Hat Enterprise Linux 8

Does not included supported version of Docker!

SIEMENS

Ingenuity for life



Red Hat



Containers at Siemens

Today and Tomorrow

05.12.2019

10–18 Uhr

Siemens und Red Hat laden ein.

DEC



Leonardo Hotel
Munich City East

Carl-Wery-Str. 39
München-Neuperlach

Was erwartet euch?

- Serviceangebot des EAGLE Data Centers
- Alles rund um Kubernetes
- Jede Menge Kunden-/Projektberichte
- Netzwerken bei Drinks & Snacks

Wer sollte teilnehmen?

Software-Entwickler/-Architekten, Admins,
DevOps, Applikations-/Lösungsanwender,
Verantwortliche/Leiter für Digitalisierung
und IT.

More
information on
Yammer!



Die Teilnahme ist kostenfrei, Anmeldung erwünscht:
<https://red.ht/34emLG9>



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

