



# **Optimizing for Production Workloads**

Dan Walsh Red Hat @rhatdan

Samuel Ortiz @sameo



Core OS



redhat.®

PDF

PDF

Linux

PDF

Linux

Containers

! \$ \*  
Swear  
Jar  
# #





# What do you need to run a container

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





# 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](https://github.com/containers/image)







# What do you need to run a container

- Standard Definition of what a container image is
  - OCI Image Bundle Definition
- Mechanism to pull images from a container registry to the host
  - [github.com/containers/image](https://github.com/containers/image)
- Ability to explode images onto COW file systems on disk
  - [github.com/containers/storage](https://github.com/containers/storage)





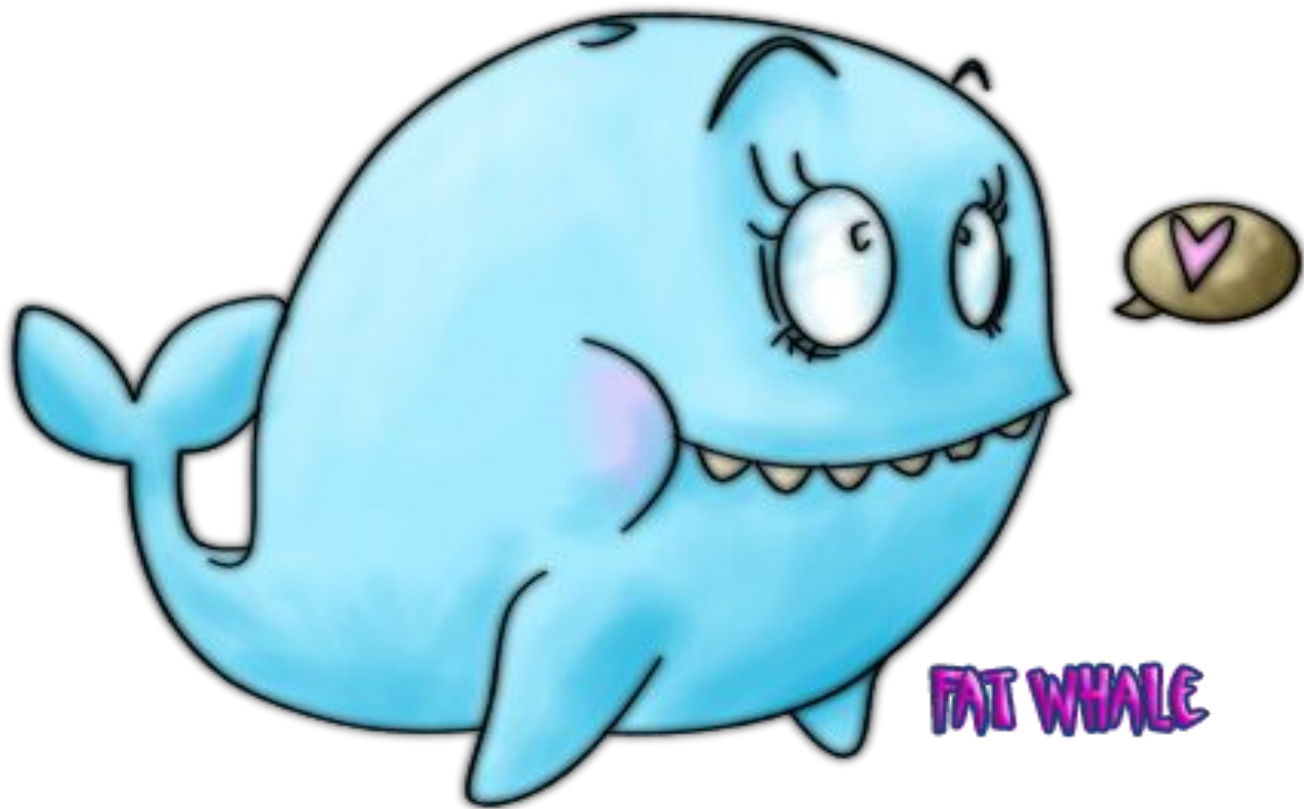
# What do you need to run a container

- Standard Definition of what a container image is
  - OCI Image Bundle Definition
- Mechanism to pull images from a container registry to the host
  - [github.com/containers/image](https://github.com/containers/image)
- Ability to explode images onto COW file systems on disk
  - [github.com/containers/storage](https://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





# What does OpenShift/Kubernetes need to run a container?

CRI - Container Runtime Interface



#nbigfatdaemons



# What does Kubernetes need to run a container?

CRI - Container Runtime Interface



Core OS

Kubernetes tells CRI to run Container Image:



# What does Kubernetes need to run a container?

CRI - Container Runtime Interface



Kubernetes tells CRI to run Container Image:

- CRI needs to pull image from Container Registry



# What does Kubernetes need to 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



# What does Kubernetes need to 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





# 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”

**#nbigfatdaemons**



**cri-o**



**kubernetes**

**#nobigfatdaemons**



MESOSPHERE

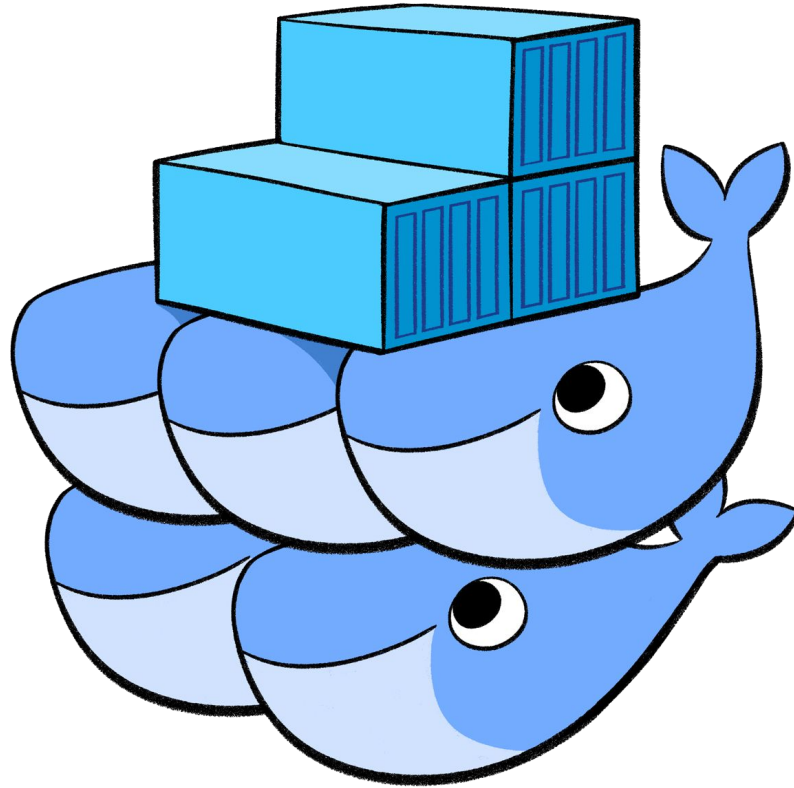
#nbigfatdaemons



#nobigfatdaemons



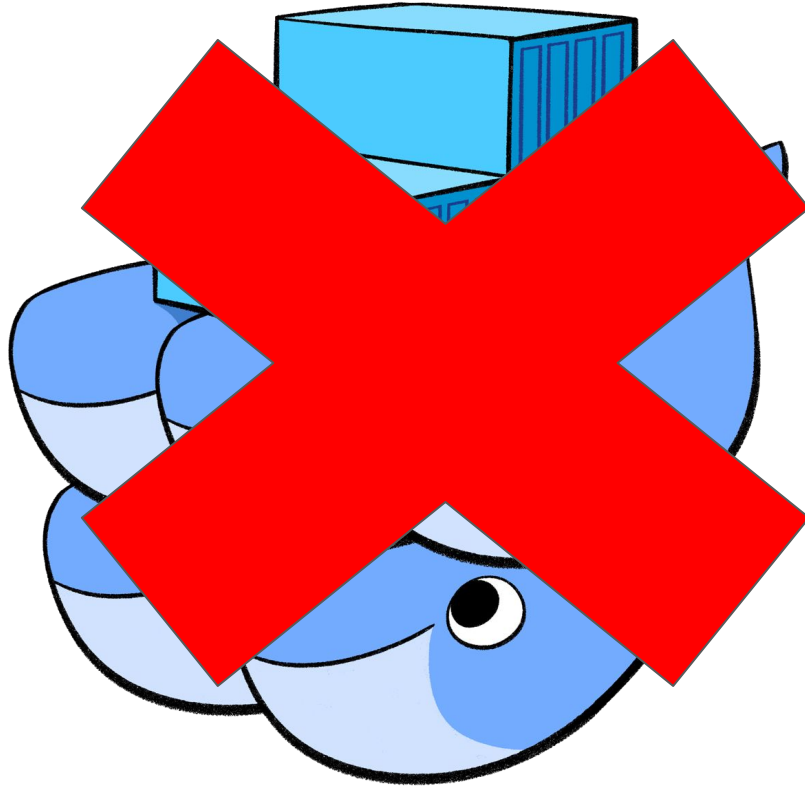
S  
W  
A  
R  
M



#nobigfatdaemons



S  
W  
A  
R  
M



#nobigfatdaemons



**VS.**

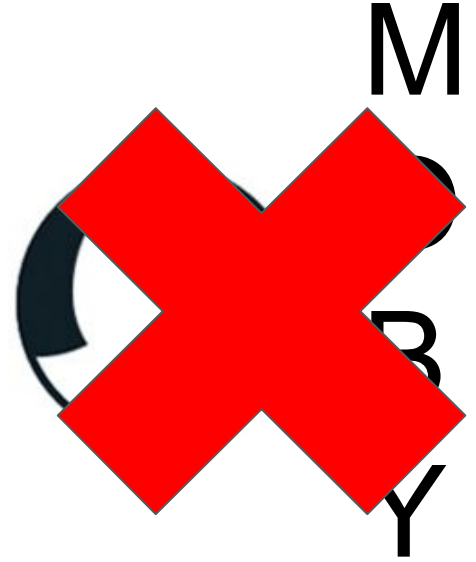


M  
O  
B  
Y

**#nobigfatdaemons**

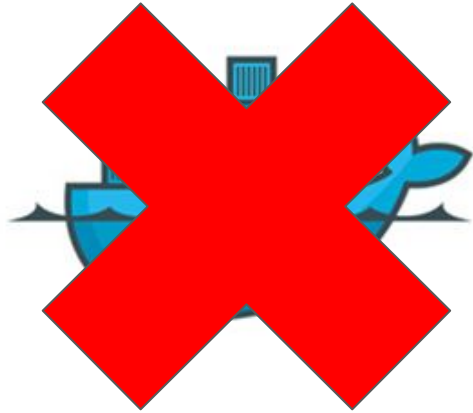


**VS.**

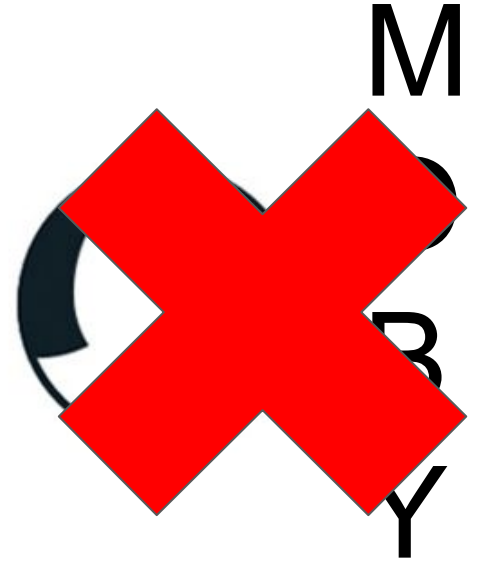


**#nobigfatdaemons**





**VS.**



**#nobigfatdaemons**



**cri-o**



**kubernetes**

**#nobigfatdaemons**



# Overview of additional components

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



# 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





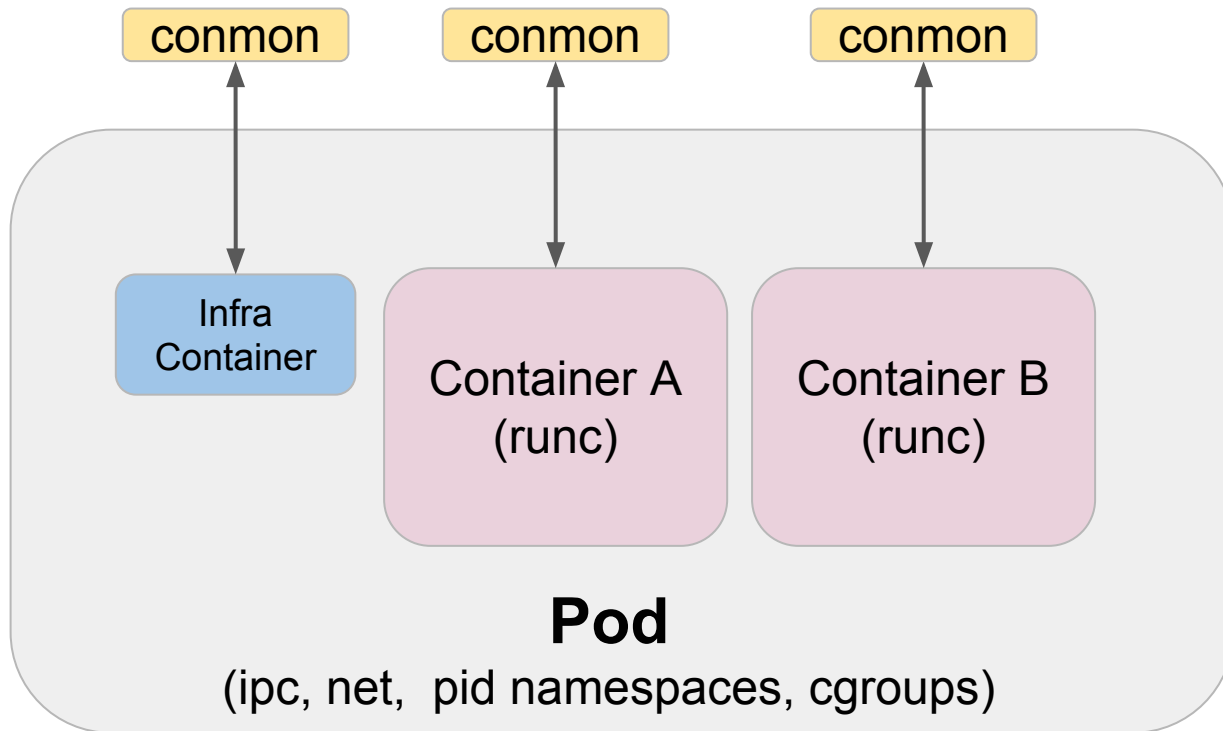
# Overview of additional components

- **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



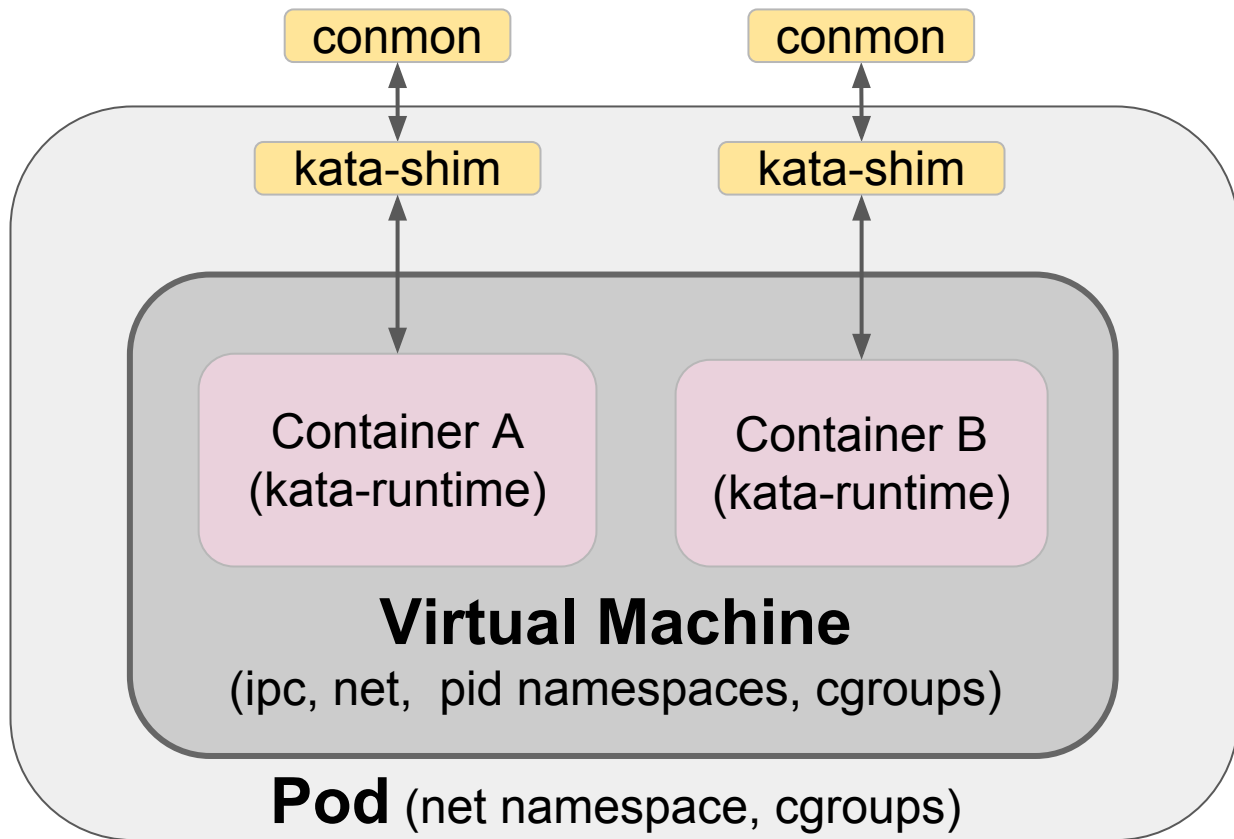


# Pod architecture (runc)



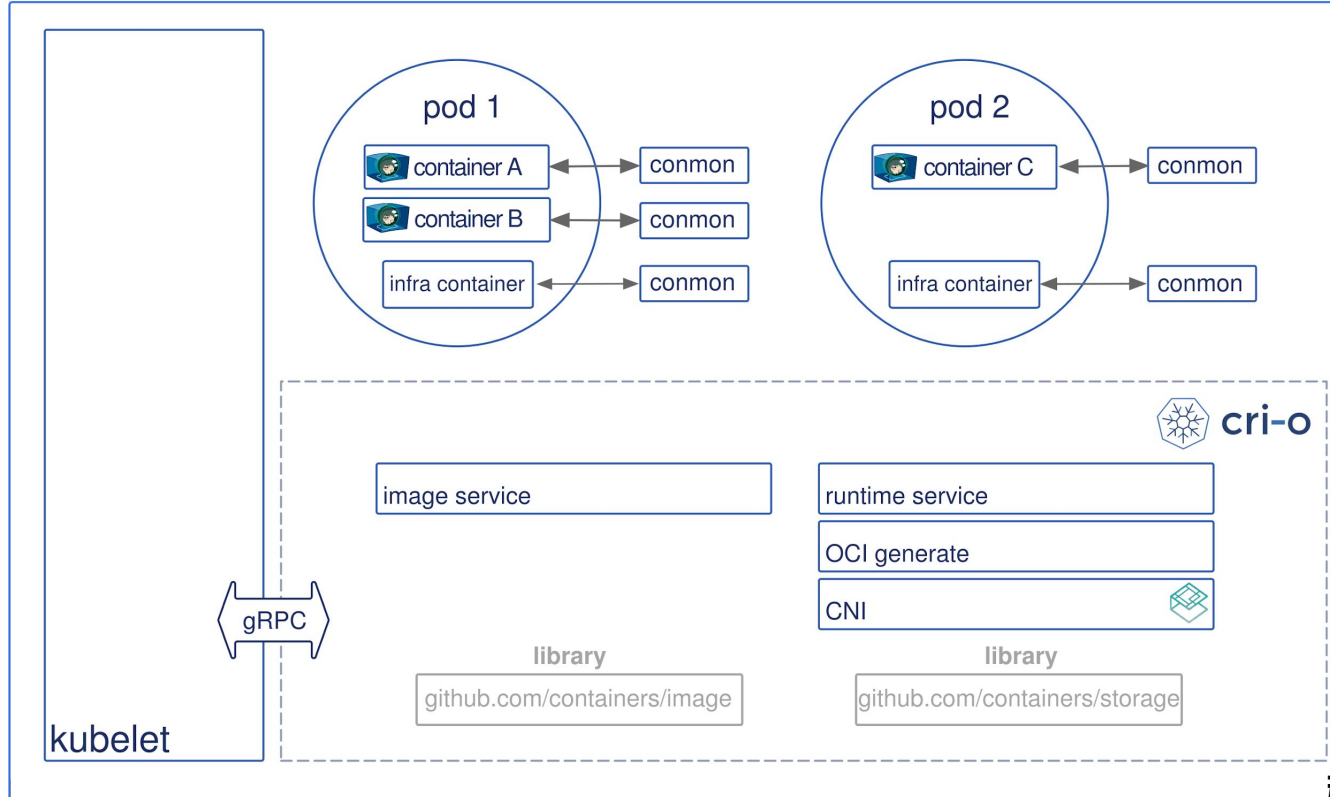


# Pod architecture (Kata Containers)





# Architecture



#nobigfatdaemons





# Status

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



# Status

- **All** e2e, cri-tools, integration (>500) tests passing.
  - **No PRs merged without passing all the tests.**
- 1.0.7 (kube 1.7.x) supported.
  - Currently available as tech preview in Openshift 3.7 on RHEL, Tech Preview



# Status

- **All e2e, cri-tools, integration (>500) tests passing.**
  - **No PRs merged without passing all the tests.**
- **1.0.7 (kube 1.7.x) supported.**
  - Currently available as tech preview in Openshift 3.7 on RHEL, Tech Preview
- **1.8.4 (kube 1.8.x) supported.**
  - Available for OpenShift origin 3.8.
  - Running on Openshift Online Now

**#nbigfatdaemons**



# Status

- **All e2e, cri-tools, integration (>500) tests passing.**
  - **No PRs merged without passing all the tests.**
- **1.0.7 (kube 1.7.x) supported.**
  - Currently available as tech preview in OpenShift 3.7 on RHEL, Tech Preview
- **1.8.4 (kube 1.8.x) supported.**
  - Available for OpenShift origin 3.8.
  - Running on OpenShift Online Now
- **1.9.1-1 (kube 1.9.x) released.**
  - CRI-O will be fully supported in OpenShift 3.9 along with docker.



# Status

- **All e2e, cri-tools, integration (>500) tests passing.**
  - **No PRs merged without passing all the tests.**
- **1.0.7 (kube 1.7.x) supported.**
  - Currently available as tech preview in OpenShift 3.7 on RHEL, Tech Preview
- **1.8.4 (kube 1.8.x) supported.**
  - Available for OpenShift origin 3.8.
  - Running on OpenShift Online Now
- **1.9.1-1 (kube 1.9.x) released.**
  - CRI-O will be fully supported in OpenShift 3.9 along with docker.
- **Goal for OpenShift 3.10 is to fully support CRI-O by default.**



# Status

- **All** e2e, cri-tools, integration (>500) tests passing.
  - **No PRs merged without passing all the tests.**
- 1.0.7 (kube 1.7.x) supported.
  - Currently available as tech preview in OpenShift 3.7 on RHEL, Tech Preview
- 1.8.4 (kube 1.8.x) supported.
  - Available for OpenShift origin 3.8.
  - Running on OpenShift Online Now
- 1.9.1-1 (kube 1.9.x) released.
  - CRI-O will be fully supported in OpenShift 3.9 along with docker.
- Goal for OpenShift 3.10 is to fully support CRI-O by default.
- Maintainers/contributors from **Red Hat, Intel, Lyft, SUSE and many others.**

**#nobigfatdaemons**



## Status

**CRI-O is now powering nodes on OpenShift Online.**

**#nbigfatdaemons**



" CRI-0 just works for them,  
so they haven't had much to say"

**#nbigfatdaemons**

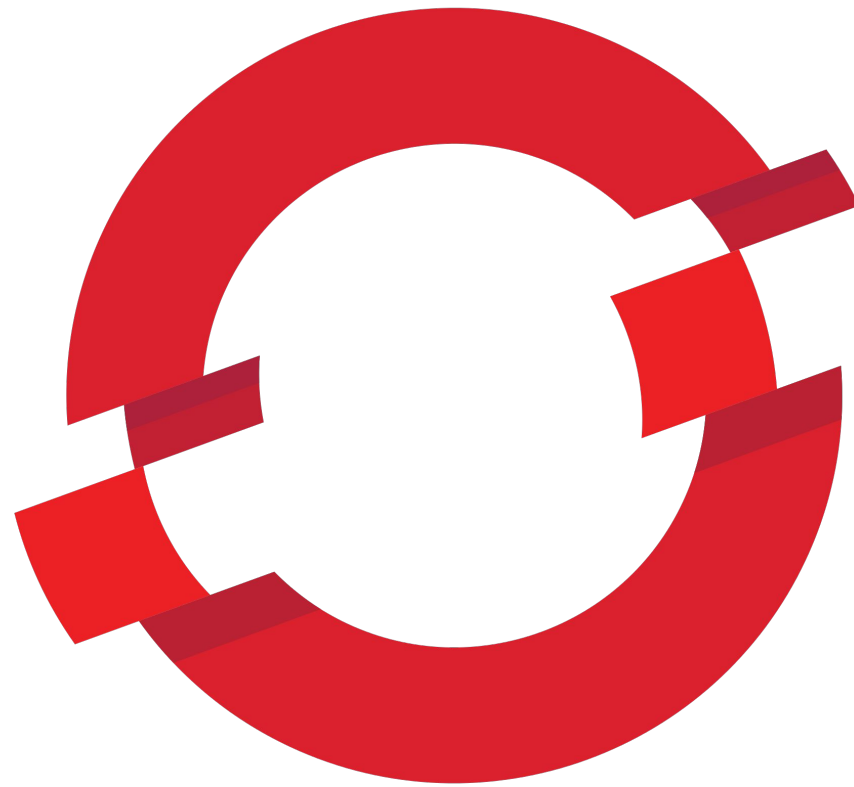




Making running containers in production

**boring**

#nbigfatdaemons



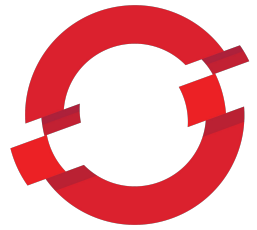
**OPENS SHIFT**

**#nobigfatdaemons**



## What else does OpenShift need?

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



**OPENSIFT**

**#nobigfatdaemons**



**#nobigfatdaemons**



# Introducing Buildah



**buildah**

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

**#nbigfatdaemons**



# buildah

Coreutils for building containers. Simple interface

**#nobigfatdaemons**



# buildah

Coreutils for building containers. Simple interface  
# ctr=\$(buildah from fedora)

#nbigfatdaemons



# buildah

Coreutils for building containers. Simple interface

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

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

**#nobigfatdaemons**



- docker docs
  - Search the docs
  - Guides
  - Product manuals
  - Glossary
  - Reference
  - Samples
- 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
--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.

- 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
```

**#nbigfatdaemons**



# buildah

Coreutils for building containers. Simple interface

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

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

```
# cp -R src $mnt
```

```
# dnf install --installroot=$mnt httpd
```

**#nbigfatdaemons**



# 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
```

**#nbigfatdaemons**



# 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
```

**#nbigfatdaemons**



# 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
```

**#nbigfatdaemons**



# 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
```

#nbigfatdaemons



**buildah**

**Dan Wait!**

**#nobigfatdaemons**





**buildah**

**Dan Wait!**

**What about Dockerfile?????**



# 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 .`

**#nbigfatdaemons**



**buildah**

**What about other formats for the scripting language?**

**#nbigfatdaemons**



**buildah**

What about other formats for the scripting language?

**BASH**

**#nbigfatdaemons**



**buildah**

**What about other formats for the scripting language?**

**BASH**

We want others to build higher level tools on Buildah.

**#nbigfatdaemons**



# buildah

**What about other formats for the scripting language?**

## BASH

We want others to build higher level tools on Buildah.

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

**#nbigfatdaemons**



# buildah

**What about other formats for the scripting language?**

## 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.

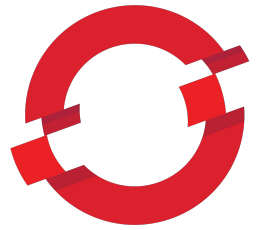
**#nbigfatdaemons**





## 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?



**OPENSIFT**

**#nobigfatdaemons**



# Introducing podman part of the libpod effort

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

<https://github.com/projectatomic/libpod>

**#nbigfatdaemons**



# Introducing podman

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

```
# podman ps -a
```

<https://github.com/projectatomic/libpod>

**#nbigfatdaemons**



# 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**



# Introducing Skopeo



<https://github.com/projectatomic/skopeo>

**#nbigfatdaemons**



# 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`





# Questions

Blog: <https://medium.com/cri-o>

Github:

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

IRC: freenode: #cri-o

IRC: freenode: #podman

Site: <https://cri-o.io>