



State of the Container Ecosystem

Dan Walsh & Mrunal Patel, Red Hat
OpenShift Commons Gathering - Austin, Texas Dec 5, 2017



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



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
- Ability to explode images onto COW file systems on disk
 - github.com/containers/storage

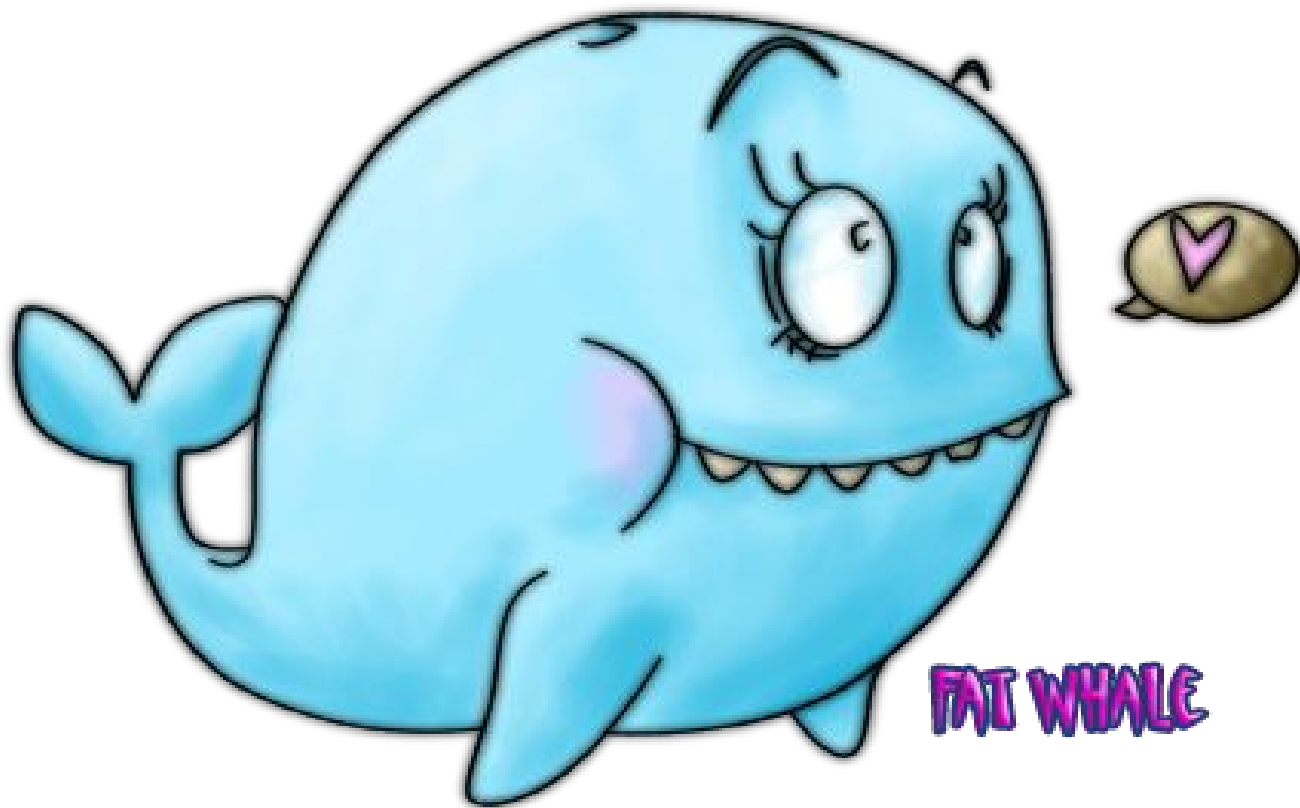


What do you need to run a container

- Standard Definition of what a container image is
 - OCI Image Bundle Definition (1.0)
- 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



FAT WHALE



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”

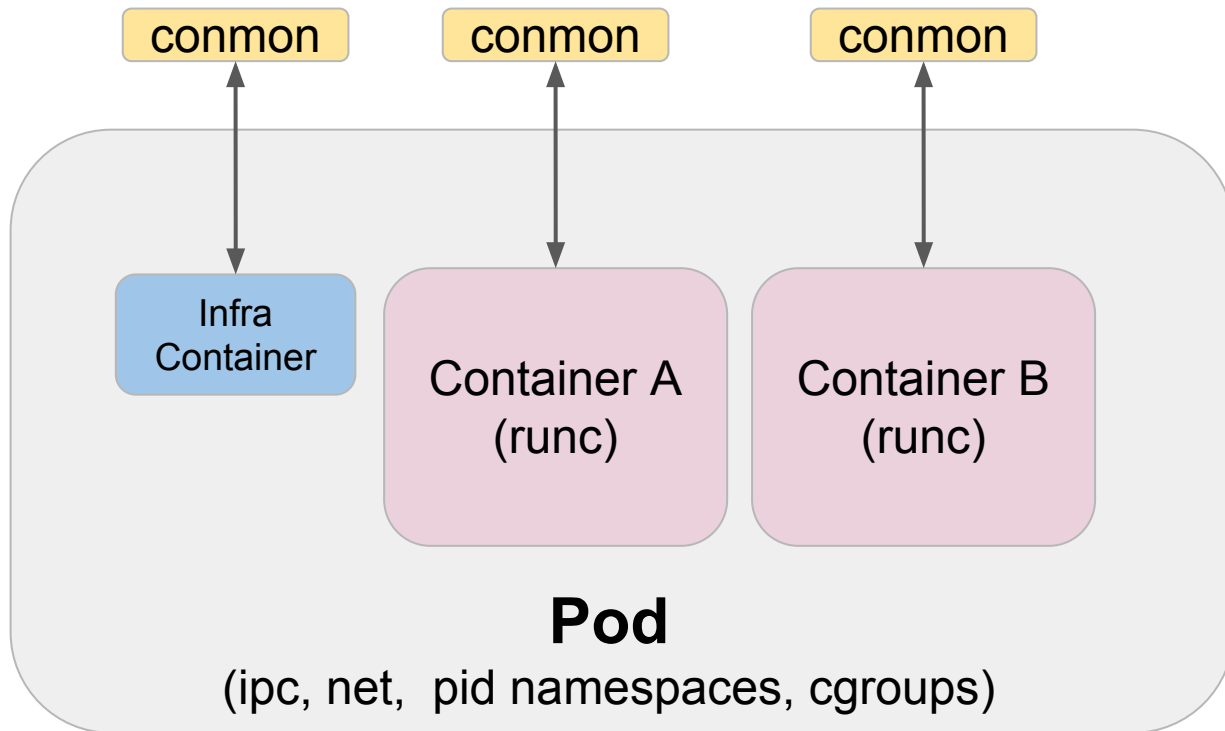


Overview of components

- **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

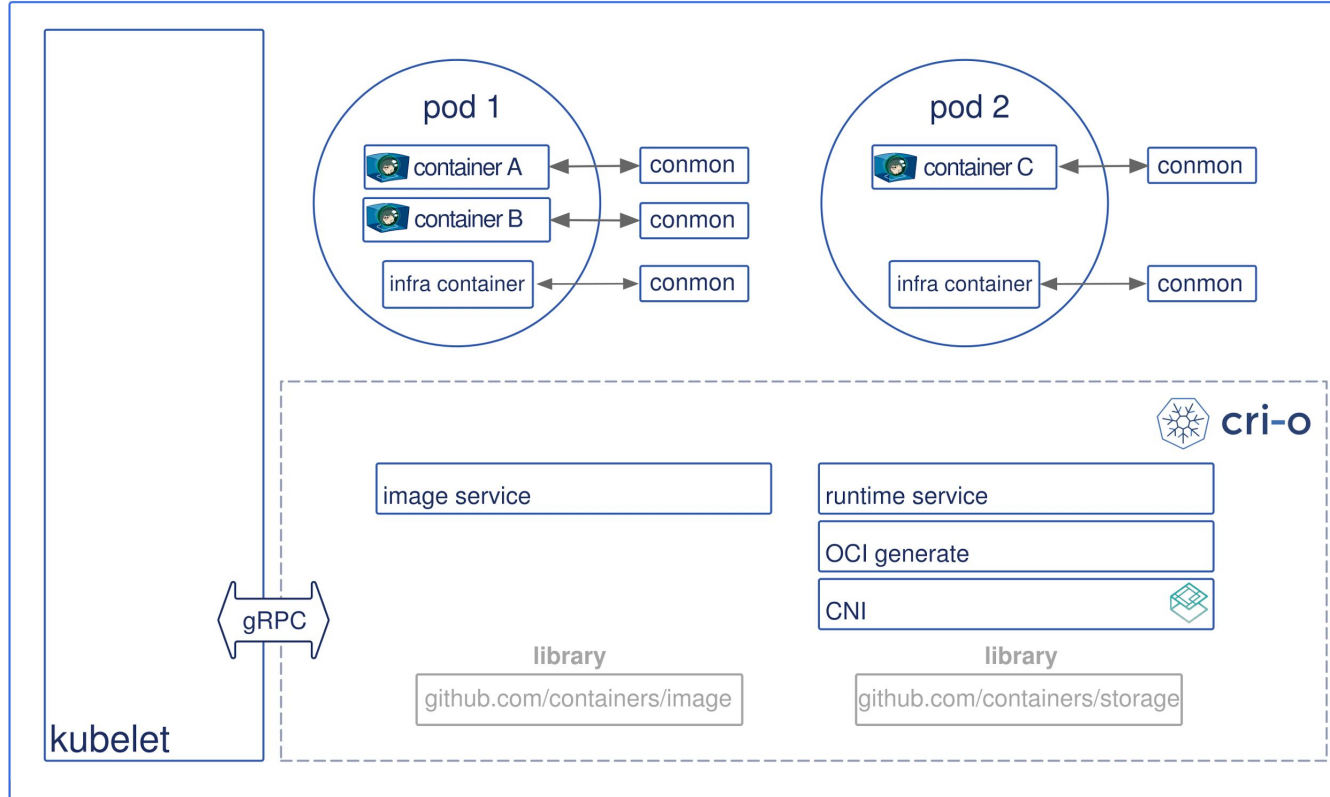


Pod architecture (runc)





Architecture





cri-o



kubernetes



Status

- **All** e2e, cri-tools, integration (>300) 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.
- 1.8.3 (kube 1.8.x) supported.
 - Available for OpenShift origin 3.8.
 - OpenShift Online targeted for next release.
- 1.9.0-beta1 (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, SUSE and many others.**



Demo (at Kubecon)

Come to the CRI-O talk at Kubecon (Wednesday 12/6 3:40 p.m.)

CRI-O: All the Runtime Kubernetes needs, and Nothing More.



What else does OpenShift need?

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



Introducing Buildah



buildah

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



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
```



buildah

#nbigfatdaemons

Buildah also supports Dockerfile
buildah build-using-dockerfile -f Dockerfile .



buildah

#nbigfatdaemons

Buildah also supports Dockerfile

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

Or for those lazy ones:

`buildah bud -f Dockerfile .`



buildah

#nbigfatdaemons

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



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?



Introducing ~~kpod~~ part of the libpod effort

Name working its way through legal

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

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



Introducing ~~kpod~~

Name working its way through legal

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

```
# kpod ps -a
```



Introducing ~~kpod~~

Name working its way through legal

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

```
# kpod ps -a
```

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




Introducing ~~kpod~~

Name working its way through legal

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

```
# kpod ps -a
```

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

```
# kpod exec -ti fedora sh
```



Introducing ~~kpod~~

Name working its way through legal

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

```
# kpod ps -a
```

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

```
# kpod exec -ti fedora sh
```

```
# kpod images
```

...



Introducing Skopeo



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



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>
- <https://github.com/containers/storage>
- <https://github.com/containers/image>

IRC: freenode: #cri-o

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