

Moving away from Docker and leaving its risks behind

OCI based projects Podman, Buildah, Skopeo & CRI-O.

Container tools landscape is changing. Why?



Since Open Container Initiative (OCI)
there are several new projects

What needs do these projects address?

What are these projects and when
should I use them?

What specific security concern does
each address?



Early concerns with Docker

Since the early days enterprise users of Docker had concerns

- Build requires a “big fat” daemon on every host
- Regression for integration with Kubernetes/OpenShift
- Build has secret handling issues
- Root/privileged concerns at runtime
- Root/privileged concerns with daemon
- Build requires a running container



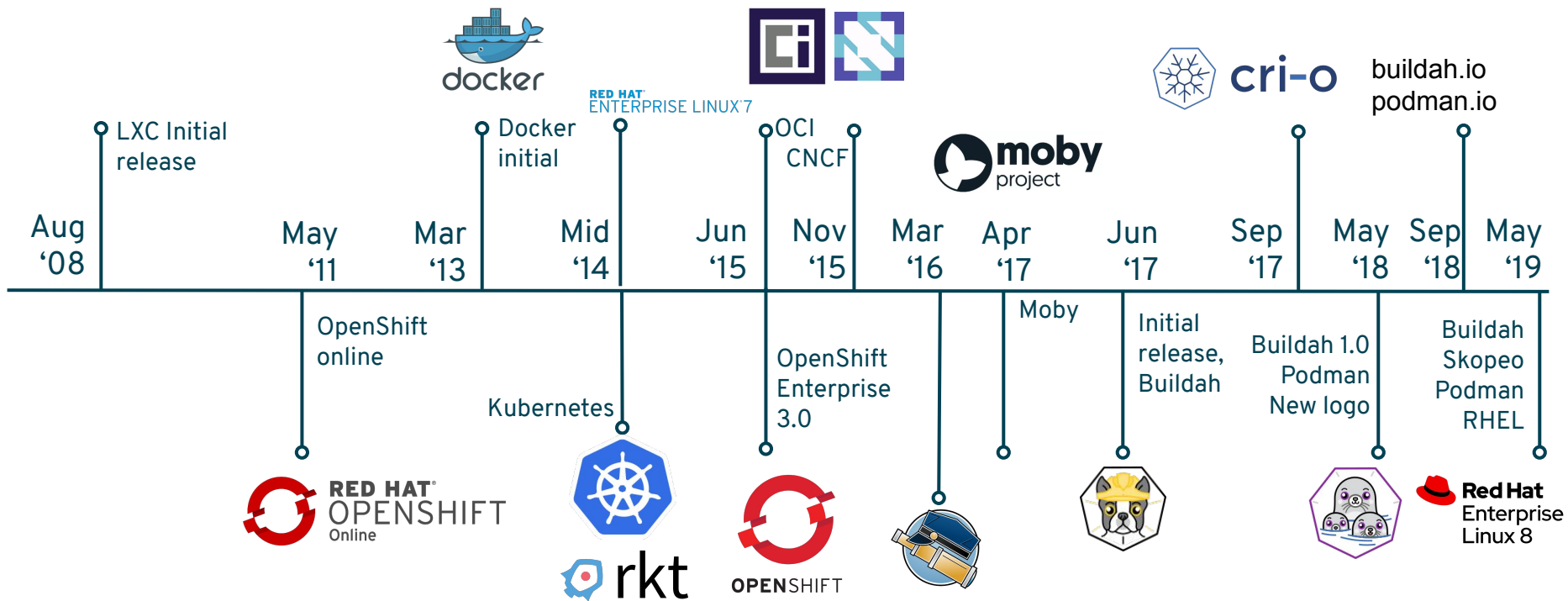


OPEN CONTAINER
INITIATIVE

- Docker, Red Hat et al. June 2015
- Two specifications
 - Image format
 - How to package an OCI Image with sufficient information to launch the application on the target platform
 - Runtime
 - How to launch a “filesystem bundle” that is unpacked on disk
- Version 1.0 of each released July 19th 2017
- Distribution spec started in April, 2018.

Containers are Linux

Container innovation continues



How did Docker change containers?

Docker Daemon

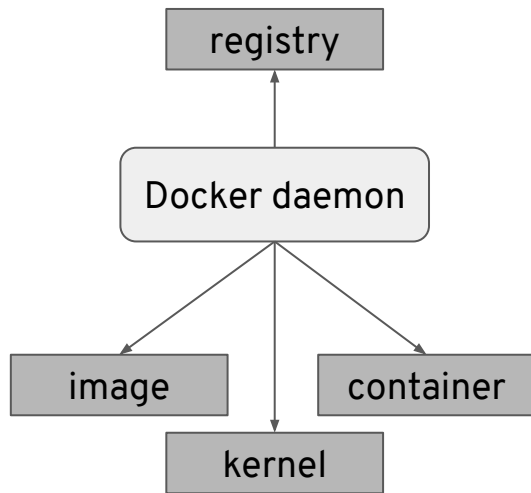
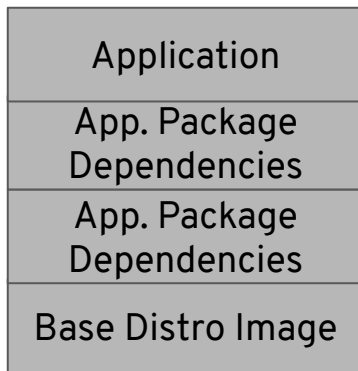
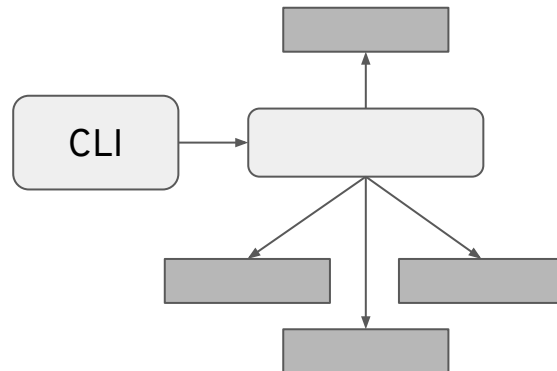


Image Layers



Docker CLI





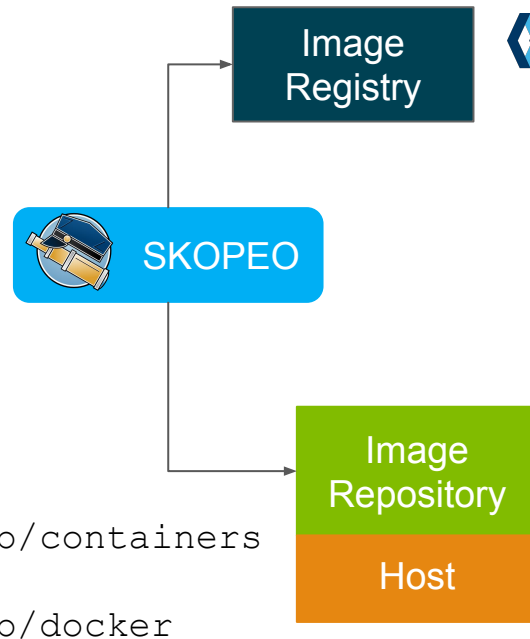
Skopeo.
The first break
away.

IMAGE COPY WITH SKOPEO

- Built for interfacing with Docker registry
- CLI for images and image registries
- Rejected by upstream Docker `_(ツ)_/`
- Allows remote inspection of image meta-data - no downloading
- Can copy from one storage to another

SECURITY FEATURES

Share securely
No daemon
Inspect remote images
No pulling potentially malicious images
Non-root copy. Bridge between registries.



QUAY





Podman.
The daemonless
client for
developers and
beyond.



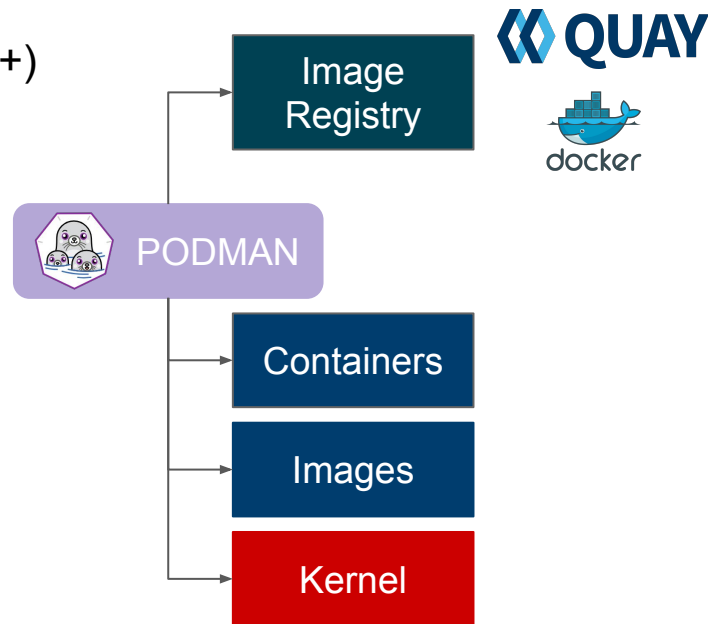
podman

The new container CLI

- @ podman.io
- Client only tool, based on the Docker CLI. (same+)
- No daemon!
- Storage for
 - Images - `containers/image`
 - Containers - `containers/storage`
- Runtime - `runc`
- Shares state with CRI-O and with Buildah!

SECURITY FEATURES

Run and develop securely
No daemon
Run without root
Isolate with user namespaces
Audit who runs what





podman

Developer's experience

- Provides a familiar command line experience compatible with the Docker CLI
- Great for running, building, and sharing containers outside of Kubernetes/OpenShift
- **Build and run containers as non-root (enhanced user namespaces)**
- Can be wired into existing infrastructure where the docker daemon/cli are used today
- Use existing Dockerfiles
- Simple command line interface, **no client-server architecture**
- Docker compatible health checks

Podman Demo



podman

But there's more : podman pod

Pods are a group of one or more containers sharing the same network, pid and ipc namespaces.

create Create a new empty pod

exists Check if a pod exists in local storage

inspect Displays a pod configuration

kill Send the specified signal or SIGKILL to containers in pod

pause Pause one or more pods

ps List pods

restart Restart one or more pods

rm Remove one or more pods

start Start one or more pods

stats Display a live stream of resource usage statistics for the containers in one or more pods

stop Stop one or more pods

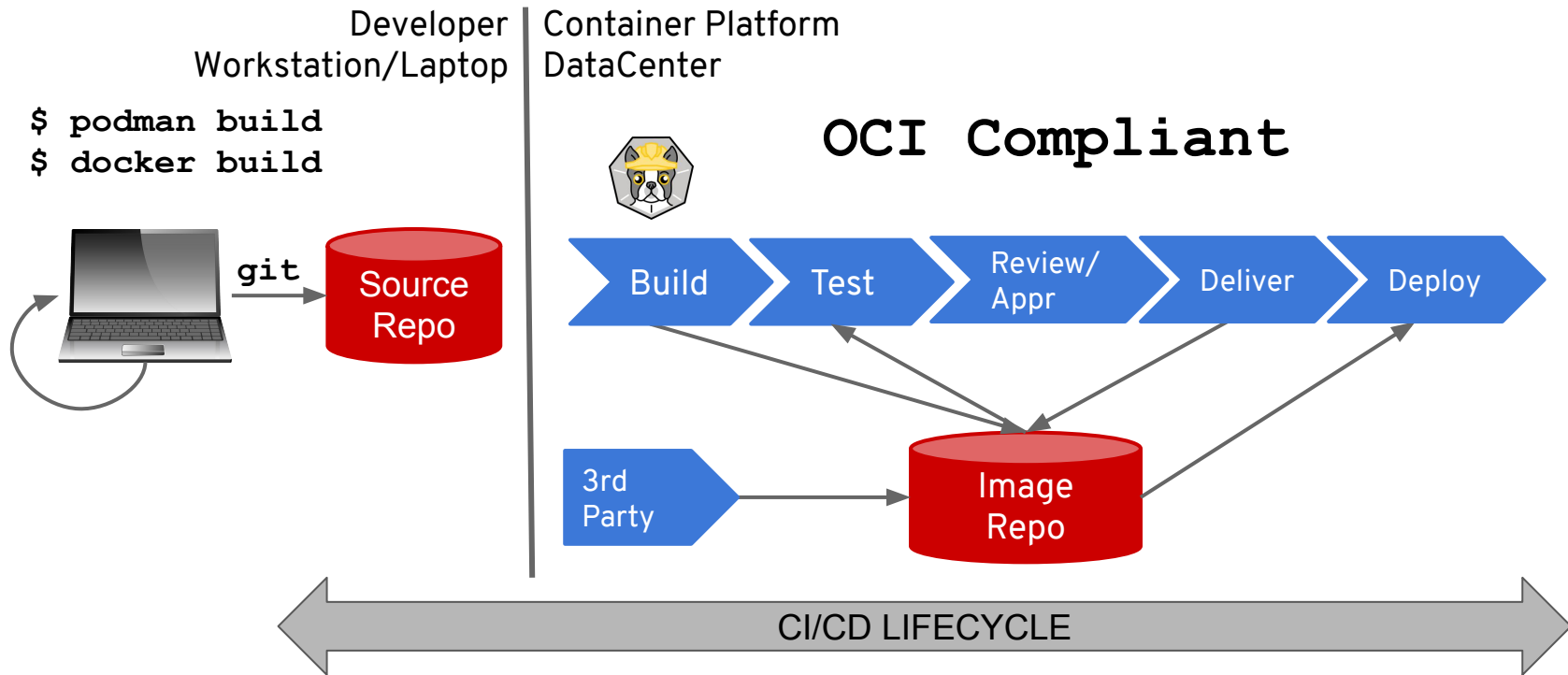
top Display the running processes of containers in a pod

unpause Unpause one or more pods



Buildah.
The secure
container
builder.

The separation of concerns





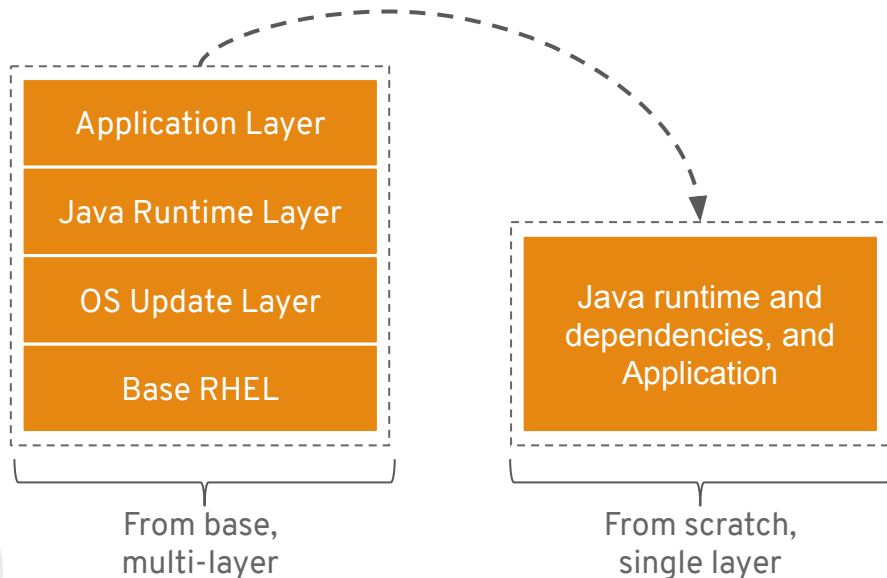
buildah

Why use Buildah?

- Now buildah.io
- Builds OCI compliant images
- No daemon - no “docker socket”
- Does not require a running container
- Can use the host’s user’s secrets.
- Single layer, from scratch images are made easy and it ensures limited manifest.
- If needed you can still maintain Dockerfile based workflow

SECURITY FEATURES

Build securely
No daemon
Shrink the attack surface
Fine-grained control of the layers
Run builds isolated
Better secret management





What does Buildah do?

buildah from - Build up a container root filesystem from an image or *scratch*.

buildah config - Adjust defaults in the image's configuration blob.

buildah run - Run a command in the container's filesystem using *runc*.

NOT like docker run. Like Dockerfile RUN.

buildah mount - Mount the container's root filesystem on the host.

buildah commit - Commit container's changes to a new image.



What does Buildah do?

buildah push - Push images to registries (Quay etc.) or a local *dockerd* instance

buildah build-using-dockerfile (a.k.a. *buildah bud*) - Build images using a Dockerfile for instructions

buildah unmount - Oh, it also unmounts container filesystems

Provide a library API that's used by the CLI

Share libraries and on-disk storage with CRI-O

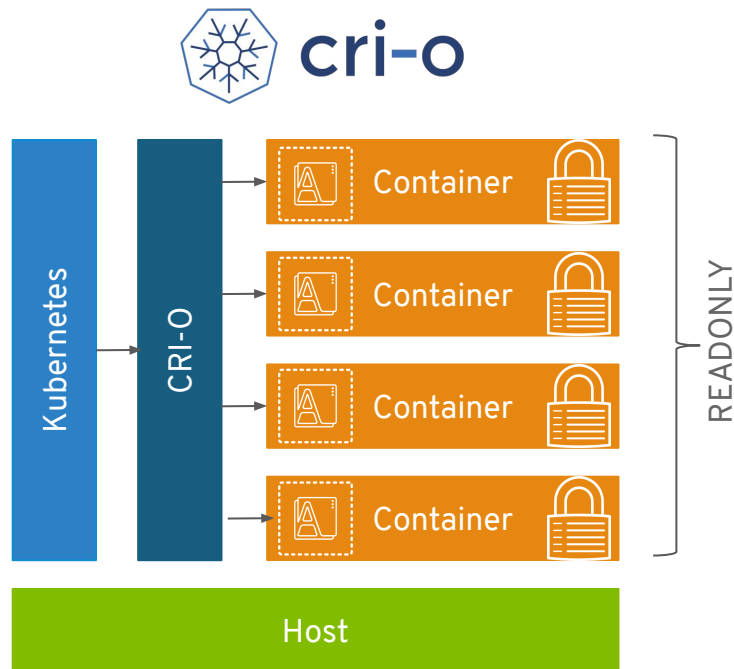


CRI-O. The OCI runtime abstraction for Kubernetes

OCI AND CRI-O



cri-o



- A Kubernetes thing
- Now part of CNCF! (April 8th)
- OCI daemon
- Implements Kubelet Container Runtime Interface (CRI)



SECURITY FEATURES

Run securely in a production cluster
No daemon
Read-only containers
Enable fewer capabilities
User namespaces
FIPS mode support



Where do I find
these projects
in Red Hat
products?

WHERE CAN I FIND THESE AWESOME PROJECTS IN RED HAT PRODUCTS?



SECURING THE PLATFORMS

Red Hat chose to move away from Docker to Podman and Buildah can CRI-O in order to provide a more secure environment for containers on both OpenShift Container Platform and RHEL



Red Hat
Enterprise
Linux 8

SMART LIGHT CONTAINERS TOOL

The container-tools package installs Podman, Buildah and Skopeo together. Users can use these in root or user namespace.



Red Hat
OpenShift
Container Platform

CONTINUOUS BUILDING, SECURE RUNTIME

Red Hat OCP has been using Skopeo for efficiency for a number of years. OCP 4 moves to using Buildah by default for building container images. CRI-O is the default container runtime. Use Podman for helping to debug pods and containers.



QUICK TIP

Learn more about RHEL 8 and OpenShift Container Platform 4 at Red Hat Summit

An abstract graphic on the left side of the slide, composed of various red and white icons. These include a cloud with a keyhole, a database cylinder, a server rack, a computer monitor, a large upward-pointing arrow, and several 'X' and 'O' symbols. Lines and dots connect these elements, suggesting a flow or process.

Open source
wins. Questions
inspire
community
innovations.



Why do we have to pull down a container just to inspect it?



Could we decouple kubernetes from the container runtime?



Is it possible to build containers on a cluster without having to install and run a daemon?



How will we design, use, debug containers on the cluster if we don't have client tools because we don't have Docker?



Getting Started

- Download Podman today
 - Package name `podman`
- It won't clash with your existing Docker
- If you feel more adventurous download Buildah too
 - Package name `buildah`
- Or all with `:container-tools`
- Lots of demos and tutorials are available
- Contribute to the projects! (Next page)



podman



buildah

Where can I learn more?



Buildah

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

<https://buildah.io/>

Podman

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

<https://podman.io/>

Skopeo

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

Other useful links

<https://github.com/opencontainers/runc>

<https://developers.redhat.com/blog/2019/02/21/podman-and-buildah-for-docker-users>

<https://www.katacoda.com/courses/containers-without-docker>

Thank you

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.



[linkedin.com/company/red-hat](https://www.linkedin.com/company/red-hat)



[facebook.com/redhatinc](https://www.facebook.com/redhatinc)



[youtube.com/user/RedHatVideos](https://www.youtube.com/user/RedHatVideos)



twitter.com/RedHat



[linkedin.com/in/williamhenry](https://www.linkedin.com/in/williamhenry)



twitter.com/ipbabble