



Red Hat Training and Certification

Managing and Building Container Images and Containers)

Travis Michette

Version 1.0

Table of Contents

1. Managing Containers with the New Runtime 1

 1.1. Deploying Containers with the New Container Runtime 1

 1.1.1. The Podman Container Engine 1

2. Podman Pods 4

3. Building Containers with Buildah 5

1. Managing Containers with the New Runtime

1.1. Deploying Containers with the New Container Runtime

1.1.1. The Podman Container Engine

RHEL8 includes the **container-tools** package module. New engine is **podman** replaces **docker** and **moby**. It also contains new tools **buildah** to build container images and **skopeo** to manage images on registries like **runc**. The new toolset allows building/running containers without daemons.

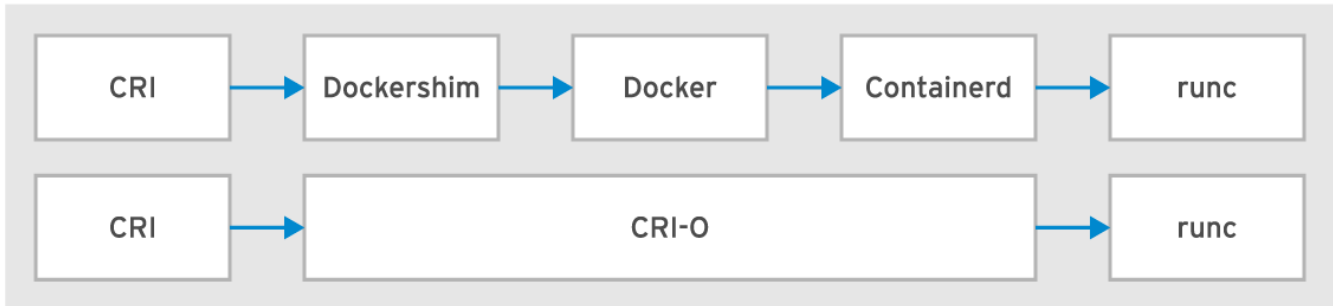


Figure 1. Docker to RHEL8 Container Runtime

Container Runtime Toolset

- Docker replaced with new container runtime
- New toolset supports OCI and reuse of third-party images
- Integrates with **audit** of Docker client-server model
- **container-tools** module provides new container runtime tools and engine.

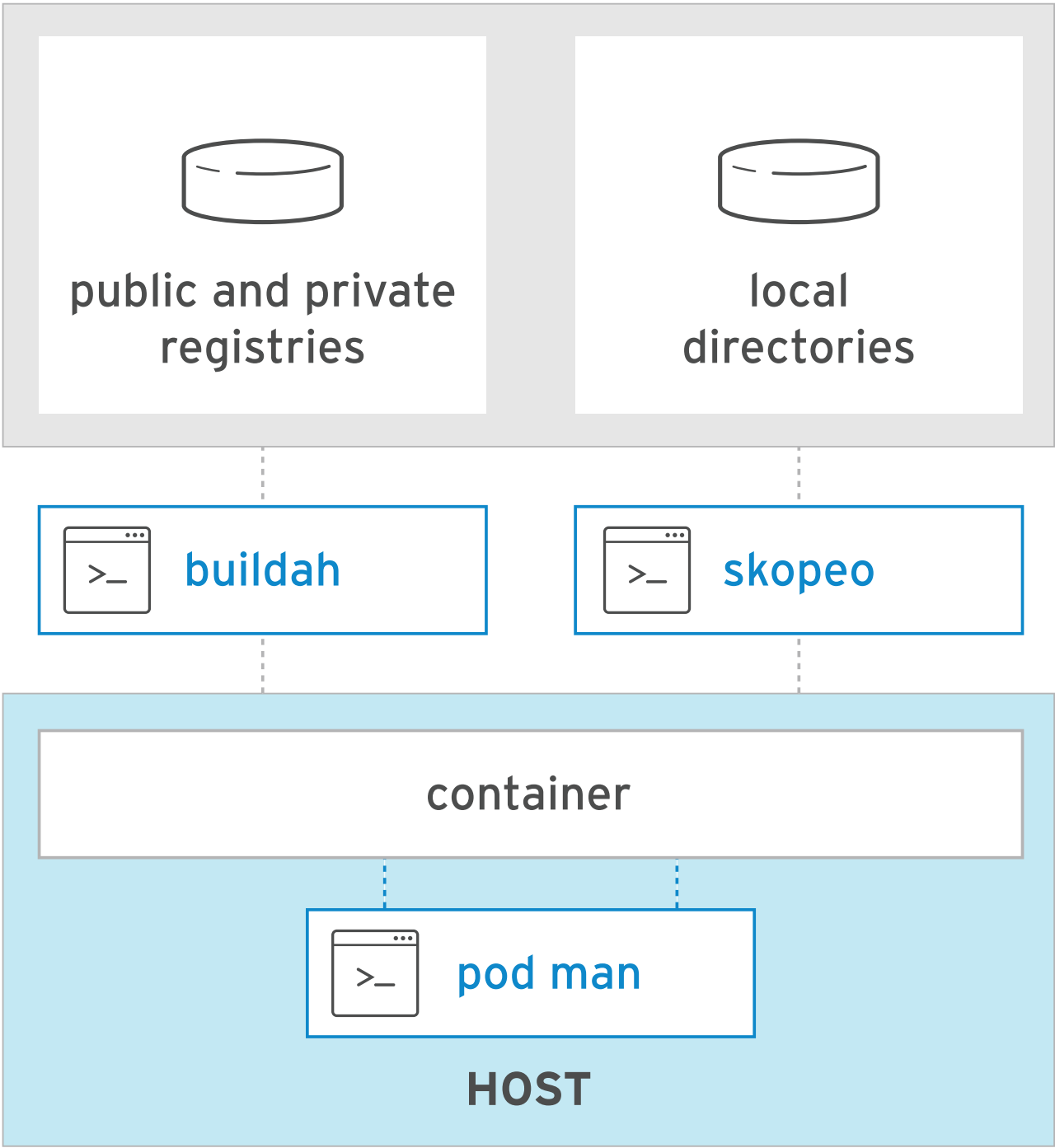


Figure 2. New Container Runtime

Describing new Container Runtime Tool

- The **podman** engine is daemonless and supporting container execution.
- **podman** syntax is similar to the docker command, supporting **Dockerfile** use
- **Buildah** builds container images, from scratch or a Dockerfile.
- Copy and inspect container images in registries with **Skopeo**
- **Skopeo** supports Docker and private registries, the Atomic registry, and local directories, including those which use OCI



RHEL8 includes **Pacemaker** containers with **podman** as a tech preview. Pacemaker supports execution of the container across multiple hosts.

Example 1. Using Container Tools

Listing 1. Installation of Container Tools

```
[student@workstation ~]$ sudo yum module install container-tools
```

2. Podman Pods

This is a test :pygments-style: tango :source-highlighter: pygments :toc: :toclevels: 7 :sectnums: :sectnumlevels: 6 :numbered:
:chapter-label: :icons: font :icons: font :imagesdir: images/

3. Building Containers with Buildah

Listing 2. Creating a Custom Container

```
[student@workstation ~]$ buildah from scratch
working-container
```

Listing 3. Naming and Inspecting a Custom Container

```
[student@workstation ~]$ buildah config --label name=My-Container working-container
[student@workstation ~]$ buildah inspect working-container
```

Listing 4. Installing Packages on Working Container

```
[student@workstation ~]$ buildah mount working-container ❶

[student@workstation ~]$ yumdownloader --destdir=/tmp redhat-release-server ❷

[student@workstation ~]$ rpm -ivh --root
/var/lib/containers/storage/overlay/a6a136063f0ada2b1ed4b01eff9a04b4d6419ae828bc4b49e742bca594e08560/merged /tmp/redhat-release-8.0-
0.39.el8.x86_64.rpm ❸

[student@workstation ~]$ cp /etc/yum.repos.d/rhel_dvd.repo
/var/lib/containers/storage/overlay/a6a136063f0ada2b1ed4b01eff9a04b4d6419ae828bc4b49e742bca594e08560/merged/etc/yum.repos.d/ ❹

[student@workstation ~]$ yum install --installroot
/var/lib/containers/storage/overlay/a6a136063f0ada2b1ed4b01eff9a04b4d6419ae828bc4b49e742bca594e08560/merged httpd ❺

[student@workstation ~]$ echo "This is a custom webserver container for me" >>
/var/lib/containers/storage/overlay/a6a136063f0ada2b1ed4b01eff9a04b4d6419ae828bc4b49e742bca594e08560/merged/var/www/html/index.html ❻

[student@workstation ~]$ yum install --installroot
/var/lib/containers/storage/overlay/a6a136063f0ada2b1ed4b01eff9a04b4d6419ae828bc4b49e742bca594e08560/merged httpd-manual ❼

[student@workstation ~]$ buildah config --cmd "/usr/sbin/httpd -DFOREGROUND" working-container ❸

[student@workstation ~]$ buildah config --port 80/tcp working-container ❹

[student@workstation ~]$ yum clean all --installroot
/var/lib/containers/storage/overlay/a6a136063f0ada2b1ed4b01eff9a04b4d6419ae828bc4b49e742bca594e08560/merged ❺

[student@workstation ~]$ buildah unmount working-container ❻

[student@workstation ~]$ buildah commit working-container my-container-image ❼

[student@workstation ~]$ buildah images ❺
```

- ❶ Mount container image filesystem for modification
- ❷ Download Red Hat Release RPM for installation
- ❸ Install Red Hat Release RPM
- ❹ Create repository for container image so files can be installed
- ❺ Install the HTTP package for a webserver

- ⑥ Create an **index.html** file for the webserver
- ⑦ Install the Apache manual for reference documentation
- ⑧ Configure webserver to run
- ⑨ Configure and open port **80** for the **TCP** protocol for the container
- ⑩ Clean up yum data to minimize required disk space
- ⑪ Unmount the container image filesystem
- ⑫ Commit the container image
- ⑬ List container images

Listing 5. Testing the Container Image

```
[student@workstation ~]$ podman run -d -p 8080:80 localhost/my-container-image

[student@workstation ~]$ curl localhost:8080
This is a custom webserver container for me

[student@workstation ~]$ curl http://localhost:8080/manual/
```

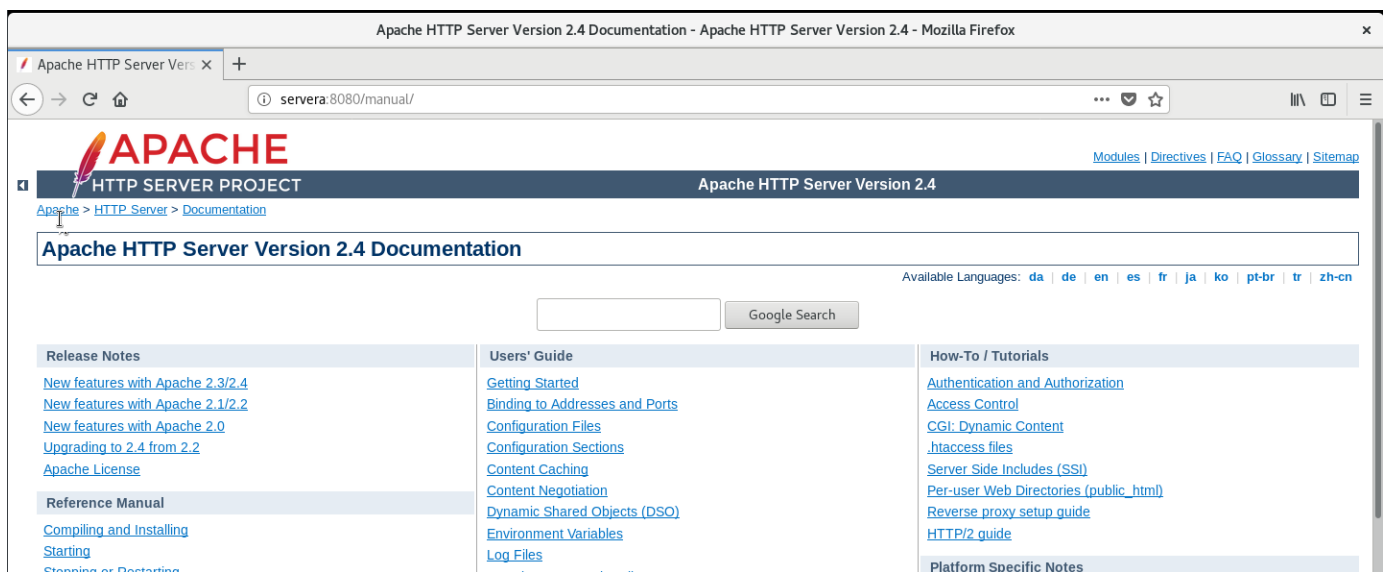


Figure 3. Testing Container

Listing 6. Stopping and Cleanup of Image

```
[student@workstation ~]$ podman list ❶
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS                NAMES
9bd572633953   localhost/my-container-image:latest /usr/sbin/httpd -...    2 seconds ago Up 1 second ago 0.0.0.0:8080->80/tcp   cranky_stonebraker

[student@workstation ~]$ podman stop 9bd572633953 ❷
9bd572633953276ac75417db3ac8e70875a0f2713e8cdfd32253fe343d06153d

[student@workstation ~]$ podman stop -a ❸

[student@workstation ~]$ podman rm cranky_stonebraker ❹

[student@workstation ~]$ podman rm 9bd572633953276ac75417db3ac8e70875a0f2713e8cdfd32253fe343d06153d ❺

[student@workstation ~]$ podman rmi localhost/my-container-image ❻

[student@workstation ~]$ buildah delete working-container ❼
```

- ❶ Listing Running Containers
- ❷ Stopping Single Container by ID
- ❸ Stopping All Running Containers
- ❹ Remove Container by Name
- ❺ Remove Container by ID
- ❻ Removing Container Image from Registry
- ❼ Delete Working Container from System