

Containers

Containers allows developers to test and build applications on any computer by putting it in a container bundled with software code, libraries, and config files

Docker

- software used to create and manage containers
- can be installed on Linux - daemon can be controlled natively
- not supported in RHEL 8

Podman

- developed by Red Hat 2018 as an alternative to docker
- daemon less, open source, linux native tool to develop, manage and run containers

Red Hat provides a set of CLI tools

- podman - directly managing pods and container images (run, stop, start, ps attach, etc)
- buildah - for building, pushing, and signing container images
- skopeo - copying, inspecting, deleting, signing images
- runc - providing container run and build features to podman and buildah
- crun - optional runtime that can be configured and give greater flexibility, control & security for root
- images - containers can be created through images and containers can be converted to images
- pods - groups of containers deployed together on the host

Basic Commands

dnf install podman -y	install podman
alias docker=podman	used when switching from doc
podman -v	check podman version
podman info	check podman environment (s if loading an image, looks loca

Image --> Run Container --> Remove it

podman search httpd	search container registries for
podman pull docker.io/library/httpd	download image from a registr
podman images	list downloaded images
podman run -d --name web1 -p 8080:80 docker.io/library/httpd	create & start a container from
podman ps	list running containers
curl localhost:8080 (or web) http://localhost:8080	verify application is accessible
podman stop web1	stop a container gracefully

podman rm -f web1	remove a container
-------------------	--------------------

Image --> Create Container --> Start Container

podman pull docker.io/library/httpd	download image from a registry
podman create --name web2 -p 8081:80 docker.io/library/httpd	create a container
podman ps -a	verify status is created -a all containers
podman start web2	start an existing container
podman ps	verify container status is running
curl localhost:8081 (or web) http://localhost:8081	verify application is accessible
podman logs web2	view output generated by container

Quadlet (Modern RHEL10) systemd container management

mkdir -p /etc/containers/systemd	create directory for quadlet definitions
nano /etc/containers/systemd/web3.container ... [Unit] Description=Apache Web Container [Container] Image=docker.io/library/httpd ContainerName=web3 PublishPort=8082:80 [Service] Restart=always [Install] WantedBy=multi-user.target	create quadlet container definition define systemd metadata description define container-specific settings specifies container image assigns name maps host port to container port defines systemd behaviour automatically restart container define boot settings allow service to start automatically
systemctl daemon-reload	reload systemd configuration
systemctl enable --now web3.service	enable service at boot and start now
systemctl status web3.service	verify service status
podman ps	verify container status is running
curl localhost:8082 (or web) http://localhost:8082	verify application is accessible