Podman Daemonless containers

CONCEPTS

> The CLI mirrors docker: alias docker=podman works for most commands. No daemon runs; each container is a child process, and it is rootless by default.

Containerfile recipe
↓ `podman build`
Image template; push to a registry
↓ `podman run`
Container running instance

> Short image names are resolved through registries.conf; use full names like docker.io/library/nginx to be explicit.

> A pod groups containers that share one network namespace, like a Kubernetes pod; they reach each other on localhost.

RUN

`podman run <image> <cmd>`
Run a command in a new container

`podman run -d --name <name> <image>`
Run in background with a name

`podman run -it <image> sh`
Interactive shell

`podman run --rm <image>`
Remove container on exit

`podman run -p <host>:<ctr> <image>`
Publish a port

`podman run -v <src>:<dst>:Z <image>`
Bind mount a path

`podman run -e <KEY>=<val> <image>`
Set an environment variable

`podman run --pod <pod> <image>`
Join an existing pod

CONTAINERS

`podman ps`
List running containers

`podman ps -a`
List all containers

`podman start <name>`
Start a stopped container

`podman stop <name>`
Stop a container

`podman rm <name>`
Remove a container

`podman logs -f <name>`
Follow container logs

`podman exec -it <name> <cmd>`
Run a command in a container

IMAGES

`podman images`
List local images

`podman pull <image>`
Download an image

`podman build -t <tag> .`
Build from a Containerfile

`podman rmi <image>`
Remove an image

PODS

`podman pod create --name <pod> -p <h>:<c>`
Create a pod with a port

VOLUMES & NETWORKS

`podman volume create <name>`
Create a named volume

SYSTEMD & QUADLET

`systemctl --user daemon-reload`
Regenerate units from Quadlet files

`systemctl --user start <name>.service`
Start a Quadlet service