Podman Daemonless containers

START HERE

`podman run -d --name <name> <image>`
Start a container

`podman ps -a`
List containers

`podman logs -f <name>`
Follow output

`podman exec -it <name> sh`
Shell inside

`podman stop <name>`
Stop it

`podman rm <name>`
Remove it

`podman images`
List images

`podman pull <image>`
Fetch an image

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

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.

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.

Rootless containers have their own storage per user and cannot bind ports below 1024 unless sysctl net.ipv4.ip_unprivileged_port_start is lowered.

RUN

`podman run <image> <cmd>`
Run a command in a new container
Add --rm to delete the container when it exits.

`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
On SELinux hosts :Z relabels the content so the container can use it.

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

`podman run --env-file <file> <image>`
Load variables from a file

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

`podman run --userns=keep-id <image>`
Map your UID into the container
Rootless: files in bind mounts keep your ownership.

`podman create <image>`
Create without starting

RUN (cont.)

`podman run --restart=always <image>`
Restart policy
Survives reboot via podman-restart.service; under systemd use Restart= instead.

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 restart <name>`
Restart a container

`podman rm <name>`
Remove a container

`podman rm -f <name>`
Force-remove, even if running

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

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

`podman inspect <name>`
Show full configuration as JSON

`podman cp <name>:<path> <dest>`
Copy files in or out

`podman stats`
Live resource usage

`podman port <name>`
Show port mappings

`podman container prune`
Remove all stopped containers

IMAGES

`podman images`
List local images

`podman pull <image>`
Download an image

`podman build -t <tag> .`
Build from a Containerfile
Also reads a Dockerfile if no Containerfile exists.

`podman build -f <file> -t <tag> .`
Build from a specific file

`podman push <image> <dest>`
Upload an image

`podman tag <image> <new>`
Add a name to an image

`podman rmi <image>`
Remove an image

IMAGES (cont.)

`podman search <term>`
Search registries

`podman login <registry>`
Authenticate to a registry

`podman image prune`
Remove dangling images

`podman save -o <file> <image>`
Export images to an archive

`podman load -i <file>`
Import images from an archive

PODS

`podman pod create --name <pod> -p <h>:<c>`
Create a pod with a port
Ports are published on the pod, not on member containers.

`podman pod ps`
List pods

`podman pod start <pod>`
Start a pod and its containers

`podman pod stop <pod>`
Stop a pod

`podman pod rm -f <pod>`
Remove a pod and its containers

VOLUMES & NETWORKS

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

`podman volume ls`
List volumes

`podman volume rm <name>`
Remove a volume

`podman network create <name>`
Create a network

`podman network ls`
List networks

`podman run --network <name> <image>`
Attach to a network
Containers on a user-defined network resolve each other by name.

SYSTEMD & QUADLET

`systemctl --user daemon-reload`
Regenerate units from Quadlet files
Run after adding or editing .container files.

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

SYSTEMD & QUADLET (cont.)

`podman quadlet install <file>`
Install a Quadlet file
Newer subcommand; absent in older 5.x releases.

`podman quadlet list`
List installed Quadlets

`loginctl enable-linger <user>`
Keep user services running after logout
Needed for rootless services to start at boot.

Quadlet file locations

Mode	Directory
rootless	`~/.config/containers/systemd/`
rootful	`/etc/containers/systemd/`

KUBE & MACHINE

`podman kube play <file>`
Run pods from Kubernetes YAML

`podman kube generate <name>`
Export a pod or container as YAML

SYSTEM

`podman info`
Show host and storage details

`podman system prune`
Remove unused data
Add --volumes to also drop unused volumes.

RECIPES

Web server with a data volume

```
podman volume create data
podman run -d --name web -p 8080:80 \
  -v data:/usr/share/nginx/html:Z \
  docker.io/library/nginx
```

Two containers in a pod

```
podman pod create --name app -p 8080:80
podman run -d --pod app --name db <db-image>
podman run -d --pod app --name web <web-image>
```

Quadlet: ~/.config/containers/systemd/web.container

```
[Container]
Image=docker.io/library/nginx
PublishPort=8080:80
[Install]
WantedBy=default.target
```

Enable that Quadlet

```
systemctl --user daemon-reload
systemctl --user start web.service
loginctl enable-linger $USER
```