

Podman Cheatsheet

This cheat sheet covers the commands used for working with Podman, a popular tool for managing containers. Podman commands are mostly compatible with Docker. As stated on the Podman landing page, "Podman is a daemonless, open source, Linux native tool designed to make it easy to find, run, build, share and deploy applications using Open Containers Initiative (OCI) Containers and Container Images."

Image repositories

```
```bash
List all local images
podman images
Remove (forced) a local image from the local cache
podman rmi -f <image>:<tag>
Pushes a container image to a remote registry
podman push <registry_url>/<username>/<image>:<tag>
Create a local container image with name and tag
podman build -t <registry_url>/<username>/<image>:<tag> .

Display historical information about a container image stored on the local machine
podman history [options] <registry_url>/<username>/<image>:<tag>

Log a user into a remote container image registry
podman login [options] <registry_url>
Log a user out of a remote container image registry
podman logout [options] <registry_url>

Pulls an image from a remote registry
podman pull [options] <remote_registry_url>/<username>/<image>:<tag>

Searches the container image registries defined in /etc/containers/registries.conf
podman search [options] <search_string>
Example of /etc/containers/registries.conf
[registries.search]
registries = ["quay.io", "registry.fedoraproject.org", "registry.access.redhat.com", "registry.centos.org",
"docker.io"]
```
```

Building images

```
```bash
Create a container image using the default Dockerfile in the local directory
podman build [options] <image>:<tag> .
create a container image using a file named Dockerfile
podman build [options] <image>:<tag> -f <Dockerfile>

Create a new tag for an existing container image in the local repository
podman tag <image_uuid> <image>:<new_tag>
podman tag <image>:<tag> <image>:<new_tag>
```
```

Containers

```
```bash
Run a container based on a given <image>:<tag> pair
podman run <image>:<tag>
Run a container in the background based on a given <image>:<tag> pair
podman run -d <image>:<tag>
Run a container with a custom name
podman run -d --name <container_name> <image>:<tag>
Run a container with a terminal and command prompt
```

```
podman run -it <image>:<tag>
Run a container that is removed after it exits
podman run --rm <image>:<tag>
Execute a command after running a container
podman run <image>:<tag> <command>
```

```
List running containers on the local machine
podman ps -a
```
```

> Note: If the image exists on the local machine, that image will be used. Otherwise, `podman run` attempts to get the container image from the remote repository specified in the command.

```
```bash
Gracefully stop a container from running
podman stop [options] <container>

Creates a container from a container image but does not start it
podman create [options] </repo/image:tag>
```

```
Restart an existing container
podman restart [options] <container>
```

```
Remove a container from the host computer
podman rm [options] <container>
```

```
Wait for the specified container to meet a condition (default is stopped)
podman wait [options] <container>
```

```
Display a live stream of a container's resource usage
podman stats [options] <container>
```
```

> Note: The command `podman stats` must be executed as `sudo` and shows only containers running with root privileges.

```
```bash
Return metadata describing a running container
podman inspect [options] <container>
```
```

Container processes and resources

```
```bash
List the containers on the local system
podman ps [options]

Create a new container image based on the current state of a running container
podman commit [options] <container> <new_image>:<tag>

Attach to a running container and views its output or controls it
podman attach [options] <container>
```
```

> Note: Use the key sequence `Ctrl+p` `Ctrl+q` to detach from the container while leaving it running.

```
```bash
Execute a command in a running container
podman exec <container> <command>

Display the running processes of a container
```

```
podman top <container>
```

```
Display the logs of a container
```

```
podman logs [options] <container>
```

```
Display the logs of a container with timestamps
```

```
podman logs -t <container>
```

```
Pause all the processes in a specified container or all containers
```

```
podman pause [options] <container>
```

```
Unpause all the processes in a specified container or all containers
```

```
podman unpause [options] <container>
```

```
List the port mappings from a container to localhost
```

```
podman port [options] <container>
```

```
```
```

Container filesystems

```
```bash
```

```
Display all the changes caused by a container to the filesystem
```

```
podman diff [options] <container>
```

```
Mount and report the location of a container's filesystem on the host computer
```

```
podman mount [options] <container>
```

```
Unmounts a container's root filesystem
```

```
podman umount [options] <container>
```

```
Export a container's filesystem to a tar file
```

```
podman export -o <output_filename> <container>
```

```
Import a tar file and saves it as a filesystem image
```

```
podman import <tar_filename>
```

```
```
```

Miscellaneous

```
```bash
```

```
Report information about the installed version of Podman
```

```
podman version
```

```
Display information about the Podman instance installed on the localhost
```

```
podman info
```

```
podman info | more -10
```

```
```
```