

Podman

Basics

Registry file: */etc/containers/registries.conf*

Login to a registry

```
podman login registry.access.redhat.com
```

Search for images

```
podman search mariadb
```

Inspect images without downloading

```
skopeo inspect docker://registry.access.redhat.com/rhsc1/mariadb-102-rhel7
```

Download images

```
podman pull registry.access.redhat.com/rhsc1/mariadb-102-rhel7
```

List local images

```
podman images
```

Start a container based on an image ID. Get the ID from docker images.

```
podman run --name apache bitnami/apache
```

control-c will stop the container for all of the run commands.

Start an image based on a tag detached

```
podman run -d --name apache bitnami/apache:2.4.52
```

Start the an app with port forwarding

```
podman run -it -p 8080:8080 bitnami/apache
```

Get running images

```
podman ps
```

Get all images

```
podman ps -all
```

Enter container in interactive shell

```
podman exec -it container-name /bin/bash
```

Commit changes to running image

```
podman commit container-name image-name
```

Check container logs

```
podman logs <Container Name>  
podman logs -f <Container Name> # Follow the logs  
podman logs --tail=25 <Container Name> # Last n lines
```

Stop a running image. The container ID will be in the `podman ps` output.

```
podman kill <Container ID>
```

Remove an image. The container ID will be in the `podman ps` output.

```
podman rm <Container ID>
```

Remove all images.

```
podman rmi --all --force
```

Export image

```
podman save image-name > /path/to/image.tar
```

Restore/Load image

```
podman load -i /path/to/image.tar
```

Parameters and Volumes

Create a container mount point

```
sudo mkdir /srv/mariadb
sudo chown -R 27:27 /srv/mariadb # UID found in podman inspect
sudo semanage fcontext -a -t container_file_t "/srv/mariadb(/.*)"
sudo restorecon -Rv /srv/mariadb
```

Run image

```
-d detached
-e per variable
-p local_port:container_port
-v local/path:/path/in/pod
```

```
podman run -d -e MYSQL_USER=user \
-e MYSQL_PASSWORD=pass -e MYSQL_DATABASE=db \
-p 33306:3306 rhsc1/mariadb-102-rhel7 \
-v /srv/mariadb:/var/lib/mysql:Z # :Z isn't needed if SELinux manually configured
```

Pods

Create a pod for rootless containers with a specific name and map ports needed. This example creates a Wordpress pod with a dedicated MySQL database using the Bitnami Wordpress image and a MySQL image from Red Hat.

Create a storage area.

```
sudo mkdir /srv/pods/wordpress/database
sudo mkdir /srv/pods/wordpress/sitedata
sudo chown -R poduser:poduser /srv/pods/wordpress # Host user running the pod
sudo semanage fcontext -a -t container_file_t "/srv/pods/wordpress(/.*)"
sudo restorecon -Rv /srv/pods/wordpress
```

Create the pod with port maps for 8443.

```
podman pod create --name press -p 8443:8443
```

Deploy the MySQL container

```
podman run -d --pod press --name mysql \
-e MYSQL_ROOT_PASSWORD=ThereIsAWordHere \
-e MYSQL_USER=wordpress \
-e MYSQL_PASSWORD=presswords \
-e MYSQL_DATABASE=wordpress \
-v /machines/pods/wordpress/database:/var/lib/mysql:Z \
mysql-80-rhel7
```

Deploy the wordpress container.

```
podman run -d --name words --pod press \
-e WORDPRESS_DATABASE_HOST=press \
-e WORDPRESS_DATABASE_USER=wordpress \
-e WORDPRESS_DATABASE_NAME=wordpress \
-e WORDPRESS_DATABASE_USER=wordpress \
-e WORDPRESS_DATABASE_PORT_NUMBER=3306 \
-e WORDPRESS_DATABASE_PASSWORD=presswords \
-v /machines/pods/wordpress/site:/bitnami/wordpress:Z \
bitnami/wordpress
```

Log in to Wordpress at <https://hostname:8443>

Systemd

Create system .service files.

To create systemd files for the above Wordpress pod:

```
podman generate systemd --files --name press
```

Creates: [container-mysql.service](#) [container-words.service](#) [pod-press.service](#)

Copy the generated file to the user's systemd directory and reload the deamons.

```
cp * ~/.config/systemd/user/
systemctl --user daemon-reload
```

Enable the service at boot time

```
systemctl --user enable pod-press.service
```

Revision #8

Created 17 February 2022 01:21:30 by Michael Cleary

Updated 8 March 2022 12:23:31 by Michael Cleary