Lab 3

Chapter 3

Outline

- Managing the Life Cycle of Containers
- Attaching Persistent Storage to Containers
- Accessing Containers

Container life cycle

Podman, implemented by the podman command, provides a set of subcommands to create and manage containers.

Developers use those subcommands to manage the container and container image life cycle.

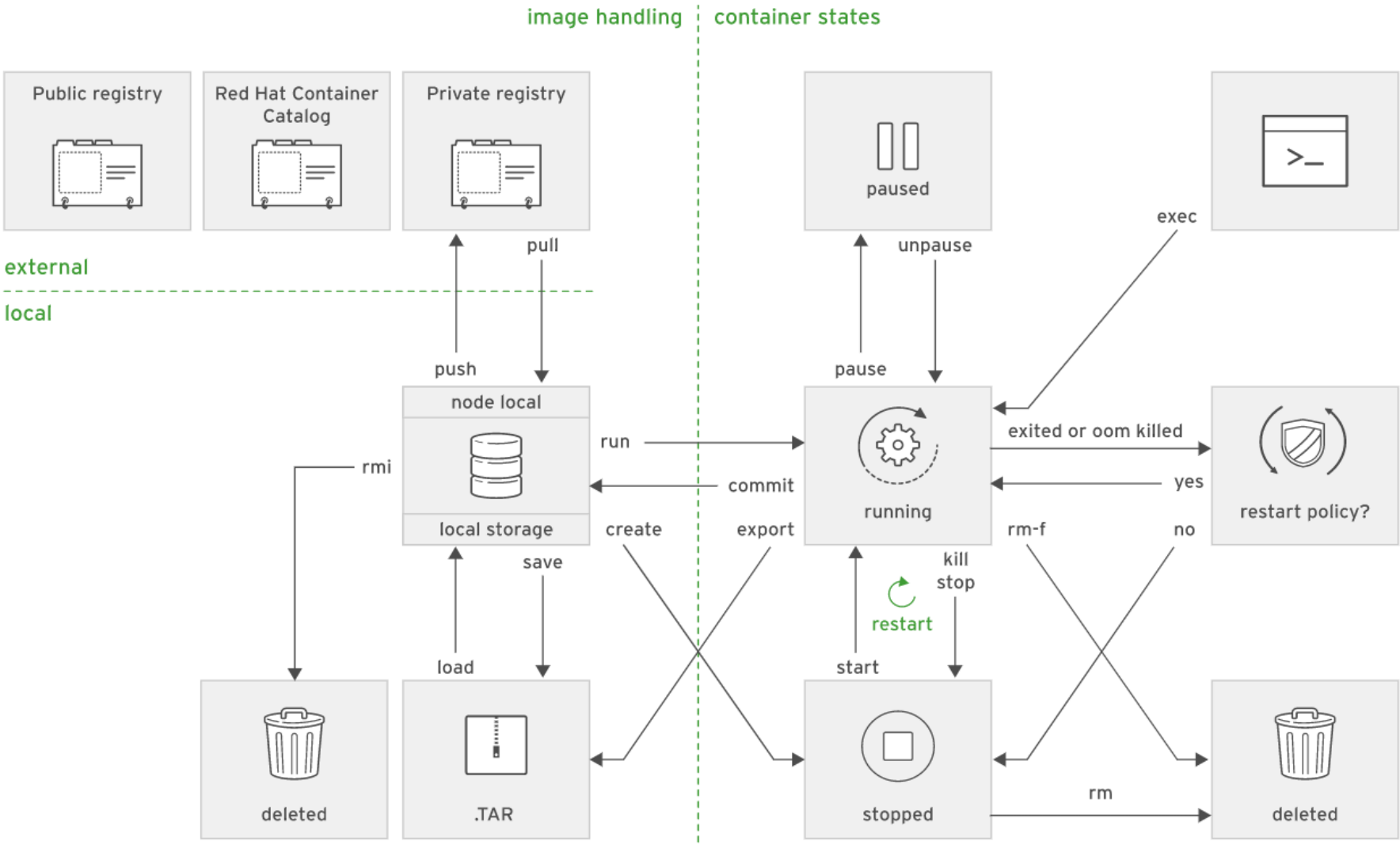


image handling
container states
Public registry
Red Hat Container Catalog
Private registry
paused
external
local
pull
push
unpause
pause
exec
node local
local storage
run
commit
rmi
create
export
save
load
running
exited or oom killed
yes
no
restart policy?
rm-f
kill
stop
restart
start
deleted
.TAR
stopped
rm
deleted

Podman Commands

- Creating Containers

```
[user@host ~]$ podman run registry.redhat.io/rhel8/httpd-24
Trying to pull registry.redhat.io/rhel8/httpd-24...
Getting image source signatures
Copying blob sha256:23113...b0be82
```

- Display containers' Ids and names

```
[user@host ~]$ podman ps
CONTAINER ID    IMAGE                 COMMAND               ... NAMES
47c9aad6049 (1)
   registry.redhat.io/rhel8/httpd-24  "/usr/bin/run-http..."  ... focused_fermat (2)
```

Podman Commands

- Define an explicit container name

```
[user@host ~]$ podman run --name my-httpd-container \
> registry.redhat.io/rhel8/httpd-24
...output omitted...AH00094: Command line: 'httpd -D FOREGROUND'
```

- Override the container entry point

```
[user@host ~]$ podman run registry.redhat.io/rhel8/httpd-24 ls /tmp
ks-script-1j4CXN
```

- Starting a bash shell

```
[user@host ~]$ podman run -it registry.redhat.io/rhel8/httpd-24 /bin/bash
 bash-4.4#
```

Running Commands in a Container

- When a container starts, it executes the entry point command. However, it may be necessary to execute other commands to manage the running container.
- The podman exec command starts an additional process inside an already running container:

```
[user@host ~]$ podman exec 7ed6e671a600 cat /etc/hostname
7ed6e671a600
```

- Podman remembers the last container created. the -l (or --latest) option:

```
[user@host ~]$ podman exec my-httpd-container cat /etc/hostname
7ed6e671a600
[user@host ~]$ podman exec -l cat /etc/hostname
7ed6e671a600
```

Managing Containers

The podman ps command displays the container ID and names for **all actively running containers:**

podman ps

```
CONTAINER ID    IMAGE                   COMMAND                 ... NAMES
47c9aad6049   rhscl/httpd-24-rhel7  "httpd -D FOREGROUND"  ... focused_fermat
```

podman ps -a

List all local containers including **the stopped ones**

Stop a container **: podman stop my-httpd-container**

Kills a container: **podman kill my-httpd-container**

Restart a container**: podman restart my-httpd-container**

Remove one container**: podman rm my-httpd-container**

Remove all containers**: podman rm -a**

Stop all containers**: podman stop -a**

Inspect a container

Show All info about the latest running container

sudo podman inspect -l

Or inspect specific container

sudo podman inspect my-httpd-container

To fetch certain info use -f

sudo podman inspect -l -f "{{.NetworkSettings.IPAddress}}"

we can pass environment variable using -e option.

***sudo podman run --name mysql-custom ***

***> -e MYSQL_USER=redhat -e MYSQL_PASSWORD=r3dh4t ***

> -d rhmap47/mysql:5.5

Guided Exercise: Managing a MySQL Container

- https://rha.ole.redhat.com/rha/app/courses/do180-4.10/52f11f0e-a277-4441-9d11-e3d56d7defca/pages/ch03s02

```
[student@workstation ~]$ lab manage-lifecycle start
```

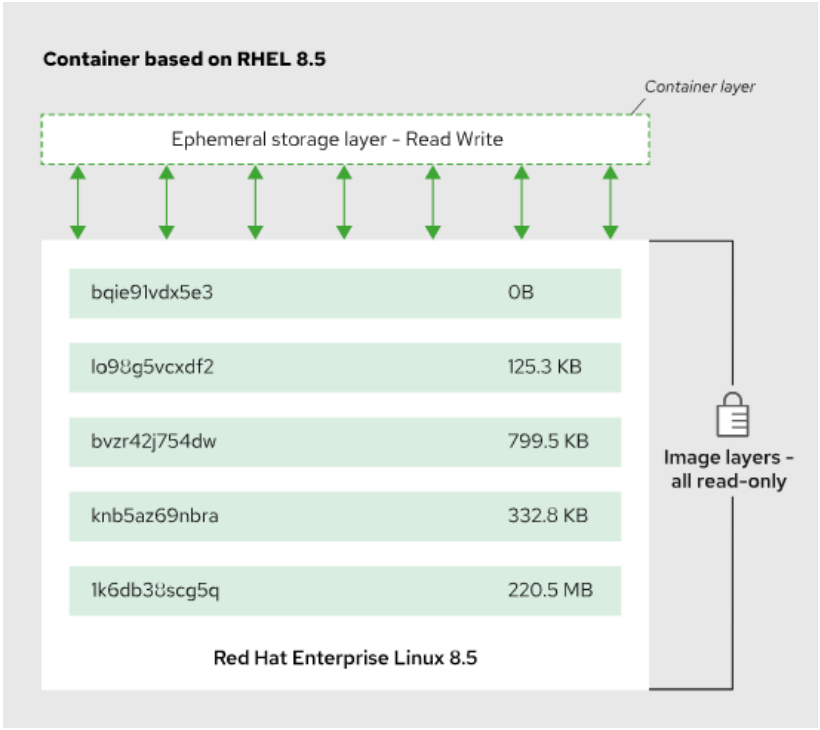
Attaching Persistent Storage to Containers(volumes)

When the container is created, temporary storage is added to manage all subsequent file changes such as log files and application data.

The problem is that if the container is destroyed e.g. scaled to zero, updated or moved to a different node, all of the written data will disappear.

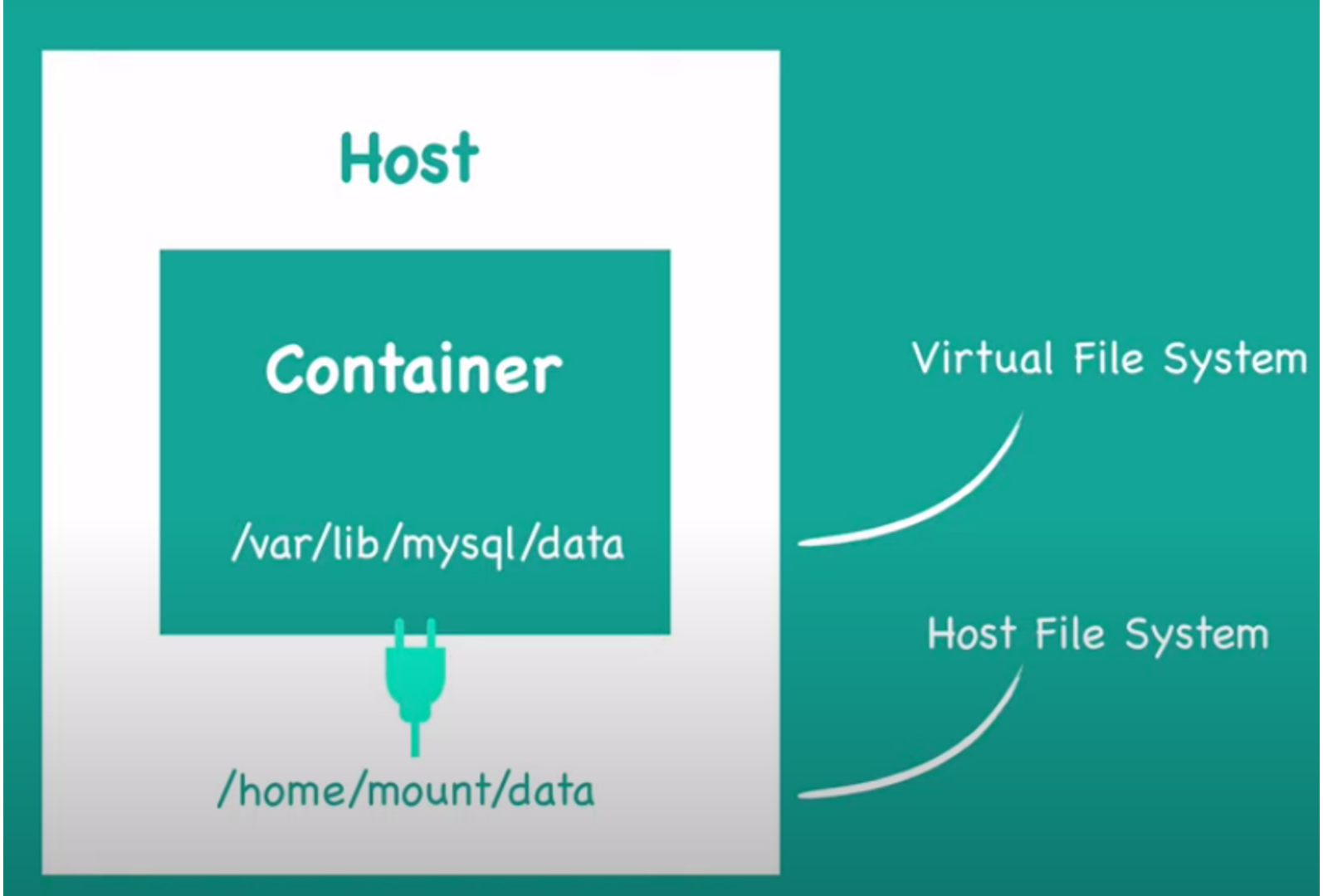
To prevent this happening, **we need to store our data in persistent storage.**

Layering


Container based on RHEL 8.5
Container layer
Ephemeral storage layer - Read Write
bqie91vdx5e3
0B
lo98g5vcxdf2
125.3 KB
bvzr42j754dw
799.5 KB
knb5az69nbra
332.8 KB
1k6db38scg5q
220.5 MB
Red Hat Enterprise Linux 8.5
Image layers - all read-only

Mounting the host directory

- Podman can mount host directories inside a running container.
- The containerized application sees these host directories as part of the container storage, much like regular applications.
- The host directory must be configured with ownership and permissions allowing access to the container.


Host
Container
/var/lib/mysql/data
/home/mount/data
Virtual File System
Host File System

Steps to mount on the host

1. **Create a directory:**
   - ✓ mkdir /home/student/dbfiles
2. **Apply the container_file_t context to the directory (and all subdirectories)**
   - ✓ podman unshare chown -R 27:27 /home/student/dbfiles
3. **Provide a session to a user id to execute commands**
   - ✓ sudo semanage fcontext -a -t container_file_t '/home/student/dbfiles(/.*)?'
4. **Apply the SELinux container policy that you set up in the first step to the newly created directory**
   - ✓ sudo restorecon -Rv /home/student/dbfiles
5. **Mounting a volume**
   - ✓ podman run -v /home/student/dbfiles:/var/lib/mysql rhmap47/mysql

Guided Exercise: Create MySQL Container with Persistent Database

- https://rha.ole.redhat.com/rha/app/courses/do180-4.10/52f11f0e-a277-4441-9d11-e3d56d7defca/pages/ch03s04

```
[student@workstation ~]$ lab manage-storage start
```

Accessing Containers

- When Podman creates containers on the same host, it assigns each container **a unique IP address** and connects them all to the same software-defined network.
- These containers can communicate freely with each other by IP address.
- Container software-defined network is only accessible from the container host.
- So we use **port forwarding** rules to allow external access to a container service using -p option.

```
[user@host ~]$ podman run -d --name apache1 -p 8080:8080 \
> registry.redhat.io/rhel8/httpd-24
```

Guided Exercise: Loading the Database

- https://rha.ole.redhat.com/rha/app/courses/do180-4.10/52f11f0e-a277-4441-9d11-e3d56d7defca/pages/ch03s06

```
[student@workstation ~]$ lab manage-networking start
```

Lab Task: Managing Containers

https://rha.ole.redhat.com/rha/app/courses/do180-4.10/52f11f0e-a277-4441-9d11-e3d56d7defca/pages/ch03s07

```
[student@workstation ~]$ lab manage-review start
```