

# podman

一款类似于 docker 的轻量级容器

```
# 安装
yum install podman

# 启动服务
systemctl enable podman.service
systemctl start podman.service

# 配置文件 /etc/containers/registries.conf
unqualified-search-registries = ["registry.access.redhat.com", "registry.redhat.io",
"docker.io"]
可以修改成
unqualified-search-registries = ["hub-mirror.c.163.com", "reg-mirror.qiniu.com", "docker.io"]

# 常用命令
podman info                -- 全局信息
podman images              -- 本地镜像列表
podman ps                  -- 已启动镜像，用 podman ps -a 查全部
podman version             -- 版本
podman search nginx        -- 搜索镜像名
podman status              -- 查看状态
podman exec -it nginx /bin/bash -- 进入容器命令行，用 podman ps 查名称

# 导入导出
podman save -o httpd-24.img imageName:tag -- 根据本地 images 制作镜像
podman load httpd-24.img -- 导入自制的镜像备份
podman export nginx -o oo.tar -- 导出镜像到 tar 包，podman ps 查名称
podman import oo.tar -- 从 tar 包导入

# 容器操作
podman start 5f8c7a923f94 -- 启动容器，通过 podman ps 来确定
podman stop 5f8c7a923f94 -- 停止容器，通过 podman ps 来确定
podman kill 5f8c7a923f94 -- 关闭容器，通过 podman ps 来确定
podman restart 5f8c7a923f94 -- 重启容器，通过 podman ps 来确定
podman rm 5f8c7a923f94 -- 删除容器，通过 podman ps -a 来确定

# 镜像操作
podman images -- 查看本地镜像
podman pull nginx -- 拉取远程镜像
podman rmi 0346349a1a64 -- 删除本地镜像，用 podman images 查询
```

```
# 生成容器并启动，以后只需要 podman start "CONTAINER ID"

-- 通过 podman images 来查询本地镜像列表
-- 参数 -p 9999:80          监听本地 9999 端口映射至容器内部的 80 端口
-- 参数 --names myNginx      容器命名为 myNginx
-- 参数 -e MYSQL_ROOT_PASSWORD=root 设置环境变量
-- 参数 -v /srv:/usr/share/nginx/html 设置挂载目录
-- 参数 -p 8888:80          宿主机的 8888 映射至容器的 80 端口
-- 参数 -P 80:8888          容器的 80 端口映射至宿主机的 8888

podman run -itd --name nginx -p 8888:80 0346349a1a64
podman run -itd --name nginx -p 8888:80 -v /srv:/usr/share/nginx/html 0346349a1a64
```

#### # 安装及启动实例

```
podman pull nginx

Resolved "nginx" as an alias (/var/cache/containers/short-name-aliases.conf)
Trying to pull docker.io/library/nginx:latest...
Getting image source signatures
Copying blob 8ddb1e6cdf34 done
Copying blob 988b92d96970 done
Copying blob b0a0cf830b12 done
Copying blob 93295add984d done
Copying blob 5252b206aac2 done
Copying blob 7102627a7a6e done
Copying blob ebde0aa1d1aa done
Copying config 7383c266ef done
Writing manifest to image destination
7383c266ef252ad70806f3072ee8e63d2a16d1e6bafa6146a2da867fc7c41759

podman images

REPOSITORY          TAG          IMAGE ID          CREATED          SIZE
docker.io/library/nginx latest        7383c266ef25     5 days ago      192 MB

podman run -itd --name nginx -p 9999:80 7383c266ef25

98a15d0536a83bfc5ba6e19ae34b2caa5c0cb146070f6250e83ddc7108caac98

# 可使用 http://192.168.198.132:9999/ 访问该服务
#
# podman run -itd --name nginx -p 9999:80 -v /srv:/usr/share/nginx/html 7383c266ef25

podman ps
CONTAINER ID   IMAGE                                PORTS          NAMES
2ec35764b947  docker.io/library/nginx:latest      0.0.0.0:9999->80/tcp  nginx

podman stop 2ec35764b947
2ec35764b947

podman start 2ec35764b947
2ec35764b947

podman restart 2ec35764b947
2ec35764b947
```

## # 安装应用实例

```
podman run -p 80:80 --name nginx \  
--privileged \  
-v /mydata/nginx/html:/usr/share/nginx/html \  
-v /mydata/nginx/logs:/var/log/nginx \  
-d nginx:1.10
```

```
podman run -p 3306:3306 --name mysql \  
--privileged \  
-v /mydata/mysql/log:/var/log/mysql \  
-v /mydata/mysql/data:/var/lib/mysql \  
-v /mydata/mysql/conf:/etc/mysql \  
-e MYSQL_ROOT_PASSWORD=root \  
-d mysql:5.7
```

```
podman run --name tomcat8080 -dit -p 8080:8080 --restart=always -v  
/data/webapps/test:/usr/local/tomcat/webapps/test tomcat
```

## # 进入 *mysql* 容器

```
podman exec -it mysql /bin/bash
```

## # 登录 *mysql*

```
mysql -proot -uroot
```

## # 查看所有数据库

```
show databases;
```