

Podman

- [Podman 介紹](#)
 - [Podman 簡介](#)
 - [Podman 安裝與設定](#)
 - [Podman 使用範例](#)
 - [Podman 與 Docker 的差異](#)
- [Podman 安裝](#)
 - [Podman 安裝](#)
 - [Podman 設定](#)



PODMAN 1 2 3 4

pod



POD?

pod 是 容器 的 集合

每个 POD 包含 一个 或多个 容器 (app)。每个 pod 包含 一个 或多个 容器 的 集合。

CRI

Container Runtime Interface

metric

pod

secret

secret。 是 一个 容器 的 集合。

Persistent Volume (PV)

持久化 存储 的 集合， 持久化 存储 的 集合， 持久化 存储 的 集合 的 集合 的 集合 的 集合。

PV 是 一个 持久化 存储 的 集合。 持久化 存储 的 集合 的 集合。

YAML

是 一个 集合。 是 一个 (HTML 的 集合) 的 集合， podman (是 docker) 的 集合 的 集合 的 集合 的 集合 Kubernetes 的 集合

YAML 的 集合 的 集合 的 集合， tab 的 集合 的 集合 space 的 集合 的 集合。 tab 的 集合 "Only one identity provider can |

YAML 在 vim 中 设置 光标列 :set cursorcolumn 设置 光标列。

```
:set cursorcolumn
```

Orchestration

编排 是指 对 容器 进行 管理 和 部署 的过程。

编排 工具 包括 Kubernetes, OpenShift, Rancher, Kubesphere 等。

ingress

ingress 是指 对 容器 进行 访问 的 入口。

ingress 控制器, 负责 管理 ingress 资源。

egress

egress 是指 容器 访问 外部 网络 的 出口。

etcd

etcd 是 分布式 数据库。kubernetes 使用 etcd 存储 数据。controller 使用 etcd 存储 数据。

etcd 使用 (manifest 文件, YAML 格式) 存储 数据。

CNCF 是 容器 编排 基金会。Kubernetes 是 CNCF 的 核心 项目。COREOS 是 CNCF 的 成员 公司。

容器 编排

容器 编排

scale-up 和 scale-out

out: 容器 编排 对 容器 进行 扩容

up: 容器 编排 对 容器 进行 扩容

statefull & stateless

statefull & stateless 2 types of services.

statefull - server

stateless - client

stateless APP server does not store any data.

log file, DB, APP log.

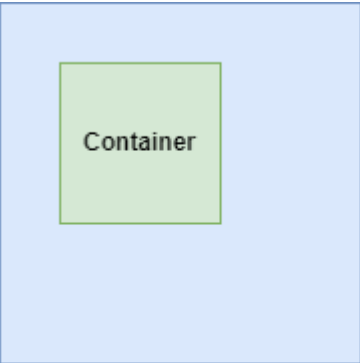
SAN ReadWriteOnce file system.

□□□□



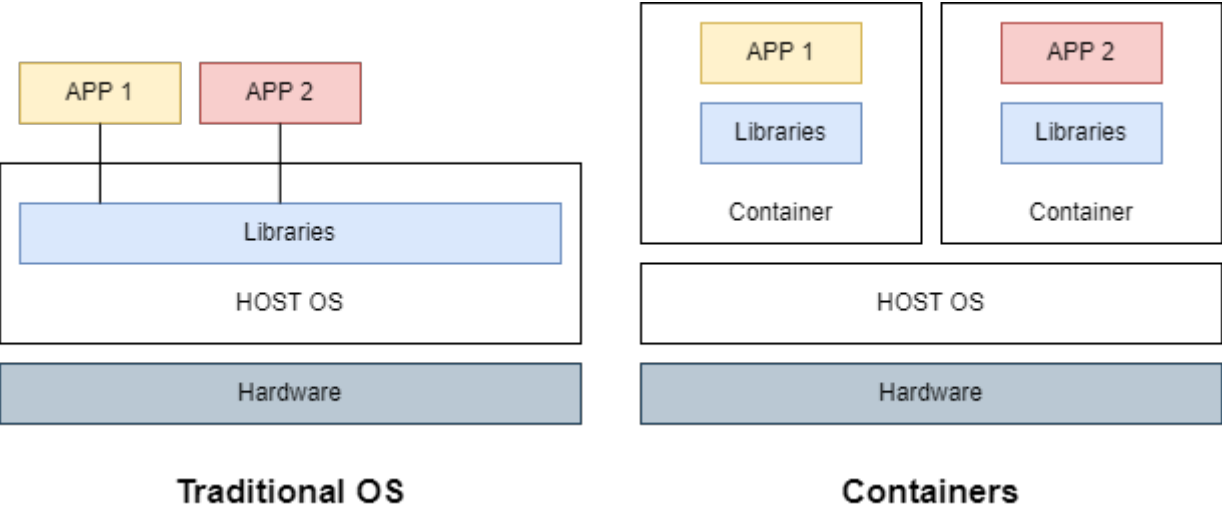
POD □ □ □ □ □ □ □ □ □ □ □ □.

□ □ □ pod □ □ □ container (app) □ □ □ □ □ □ □ □ □ □, □ □ □ pod □ □ □ □ □ □ □ □ □ □ □ □.



POD

APP □ □ □ □ □ □ □ □ □ □ OS □ □ □ □ □ □ □ □



□ □ □ □ □ □ OS □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □. □ □ □ □ APP 1 □ APP 2 □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □.

□□□

PODMAN □ □ □ □

-d

detach □ □ □ □ □ □ □ □ □ □.

-d □ □ POD □ □ □ foreground □ □ □ □ □ □ □ □, □ □ □ □ □ □ □ □ □ □. □ □ □ □ □ □ □ □ □ □ □ □.

-p 80:8080

80 : HOST OS □ □

8080 : □ □ □ □ □ □

-v □ :Z

-v □ □ □ □ □ □ :Z □ □ □ □ □ □ selinux □ □ □ □ □ □ □ □.

podman □ □ □ □ □ □

□ □ □ □ □ □ podman □ □ □ □ □ □ □ □.

podman --log-level=debug **info**

podman save

local □ □ □ □ □ □ □ □ tar □ □

podman load

tar □ □ □ □ □ □ □ □ local □ □. □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □.

podman exec pod1 /bin/bash

podman exec pod1 /bin/bash

/bin/bash pod1 (podman exec pod1 Ghost pod1)

podman **exec** -it <POD ID> /bin/bash

podman exec mariadb pod1

podman exec -it mariadb /bin/bash

mysql -u root -p

1. 背景



2. 部署环境

部署环境为 Kubernetes 集群。集群中部署了 apache、nginx、nignx、web a、web b 等应用。nginx 作为 reverse proxy，将请求转发给 apache 和 nignx。HOST port 为 8080 和 8081。

1) 1 个 POD 8080, 2 个 POD 8081

1. Docker 部署

docker 部署环境

docker 部署环境为 4 个容器，每个容器运行一个应用。

- 1. EXPOSE -p 8080 8080
- 2. EXPOSE 8080 8080
- 3. EXPOSE -p 8081 8081
- 4. -p 8081 8081

1

1. 部署环境为 Kubernetes 集群。集群中部署了 apache、nginx、nignx、web a、web b 等应用。nginx 作为 reverse proxy，将请求转发给 apache 和 nignx。HOST port 为 8080 和 8081。

2 --expose

1. 部署环境为 Kubernetes 集群。集群中部署了 apache、nginx、nignx、web a、web b 等应用。nginx 作为 reverse proxy，将请求转发给 apache 和 nignx。HOST port 为 8080 和 8081。

3 --publish-all

HOST 8080 8080, POD 8080 8080。

4 --publish (-p)

HOST . POD .

2. PODMAN

podman docker [] expose POD [] [] [] [] [] [] [].

publish rootfull podman



podman network ls

port ☐ ☐ ☐ ☐ ☐

podman port -a

POD ☐☐☐☐ ☐☐ ☐☐☐☐

□□□ □□ □□ □□□ HOST OS□ □□□□ □□ □□□□ □□ □□ □□□.

podman network reload

POD □□□□ □□□□ □□

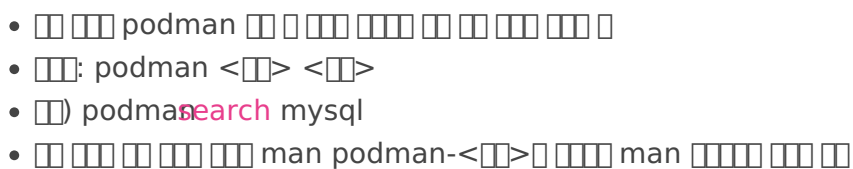
□□ □ □□□ □□□□ □□□□ □□□□□ podman □□.

podman network inspect <[] []>

```
podman network inspect podman
```

PODMAN

PODMAN 是一个 APP 可以 在 本地 运行 容器 应用



podman ps

-a 容器 ID 列表 pod 名称. 容器 pod 名称

podman **start** <POD 容器 ID>

容器 ID 列表.

podman inspect

inspect 容器 ID 列表, 返回容器元数据信息.

podman search

Registry(容器 ID) 容器名称 搜索. 容器名称 容器名称 容器名称. 容器名称 Internet 容器名称 external 容器名称.

docker 容器 library 容器名称 容器名称 容器名称. docker.io/library/<容器 ID> 容器名称 容器名称.

容器名称 容器名称 容器名称

podman **top** <POD 容器 ID>

容器名称 容器名称 容器名称

podman **images**

容器名称 tag 容器名称 容器名称

podman **pull** certbot/certbot:arm64v8-latest

podman **pull** --platform linux/arm64/v8 nginx:latest

容器名称 容器名称 容器名称

podman **rmi** <ID>

podman rmi 容器名称 容器名称 -f 容器名称 容器名称 容器名称 容器名称 容器名称 容器名称.

POD(容器) 删除

容器删除命令为 `podman rm`。删除容器时，容器内的数据将被删除。删除容器时，容器内的数据将被删除。

`podman rm <POD容器 ID>`

删除容器时，容器内的数据将被删除，使用 `-f` 强制删除容器。

`-f` 强制删除容器，使用 `podman kill` 强制删除容器。

`podman rm -f <POD容器 ID>`

PODMAN 简介



Podman 是一个容器引擎，它可以在 Linux 上运行容器。

podman <命令> <POD 名称 or ID>

Podman 支持以下命令：

run

podman run 命令用于运行容器。

- 1. pull
- 2. create
- 3. start

Podman 支持以下命令：

start

Podman 支持 start 命令，用于启动已经创建但尚未运行的容器。该命令只能在宿主机上运行，不能在容器内部运行。

restart

Podman 支持 restart 命令，用于重启已经运行的容器。

stop

Podman 支持 stop 命令，用于停止已经运行的容器。

kill

POD 1 1 kill 1 1 1 1. stop 1 1 1 1 1 1.