

Podman

- [Podman 简介](#)
 - [Podman 是什么](#)
 - [Podman 和 Docker 的区别](#)
 - [Podman 的优势](#)
 - [Podman 的架构](#)
 - [Podman 的部署](#)
- [Podman 的安装](#)
 - [Podman 的安装](#)
 - [Podman 的验证](#)



PODMAN 1 1 1 1 1 1 1

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。

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 集合(HTML 的 集合)的 集合，podman (是 docker)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。

egress

egress 是指 容器 访问 外部 网络的 出口。

etcd

etcd 是一个 分布式 数据库，kubernetes 使用 etcd 来 存储 配置 信息。

CNCF 是 云原生 基金会，Kubernetes 是 CNCF 的 核心 项目。

容器 编排

容器 编排

scale-up 和 scale-out

out: 容器 编排 和 容器 编排
up: 容器 编排 和 容器 编排

statefull & stateless

statefull & stateless 2 types of services.

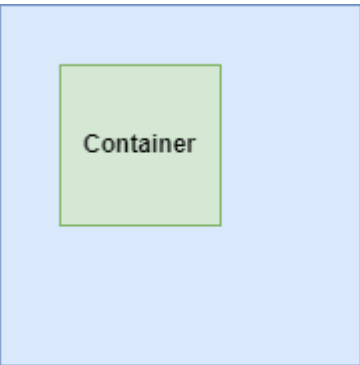
statefull - server

stateless - client

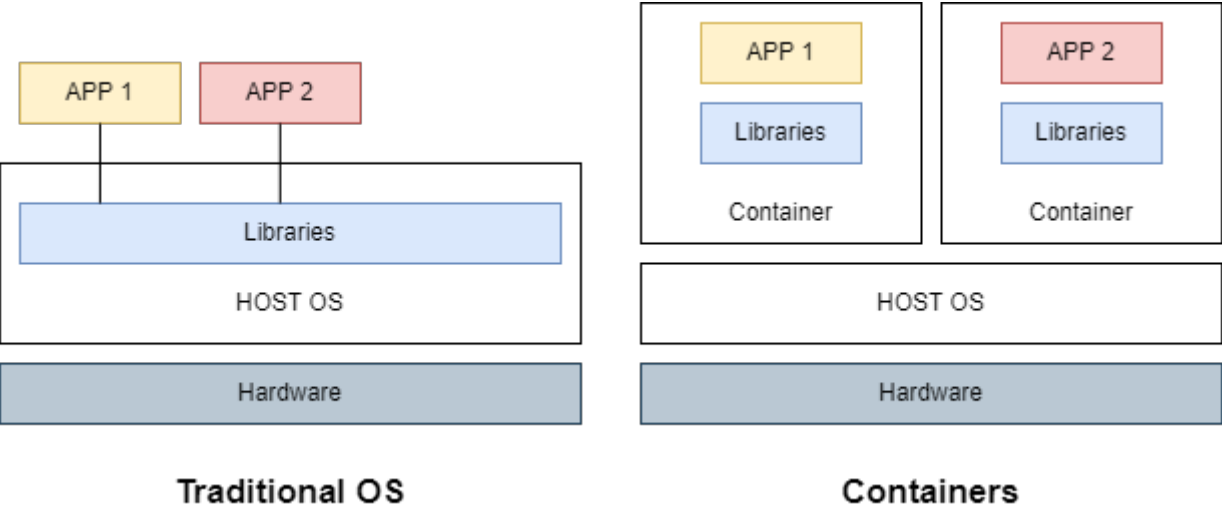
stateless APP server does not store any data.

log file, DB, APP log.

SAN ReadWriteOnce file system.

[illegible]

APP  OS 



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

APP가 실행되는 OS가 실행되는 OS와 같은 환경을 제공한다.

이러한 환경을 제공하는 APP를 컨테이너(POD)라고 하며, 이는 VM과 유사하지만 VM보다 가볍고 빠르게 실행된다.

Virtualization과 containerization의 차이

가상화와 컨테이너화는 모두 하드웨어를 추상화하여 애플리케이션을 실행시키는 기술이지만, VM은 완전한 OS를 실행시키며, 컨테이너는 호스트 OS 위에서 실행된다.

Rootful과 rootless 컨테이너의 차이

Rootful 컨테이너는 root 권한을 가지며, rootless 컨테이너는 root 권한 없이 실행된다.

이러한 차이점은 컨테이너의 실행 방식과 보안에 영향을 미친다.

- POD가 Rootful 컨테이너일 때, 이 컨테이너는 rootless 컨테이너와 다르다
- HOST OS가 1~1023번의 컨테이너를 Rootful 컨테이너로 실행한다
- POD가 Rootful 컨테이너일 때

□□□

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，将流量转发到 apache 和 web a。

集群中部署了 1 个 POD 8080，2 个 POD 8081。

1. Docker 部署

docker 部署环境

docker 部署环境为 4 个节点，每个节点部署了 4 个容器。

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

1

部署环境为 Kubernetes 集群。集群中部署了 POD 8080 和 POD 8081。POD 8080 部署了 4 个容器，POD 8081 部署了 2 个容器。

2 --expose

部署环境为 Kubernetes 集群。集群中部署了 POD 8080 和 POD 8081。POD 8080 部署了 4 个容器，POD 8081 部署了 2 个容器。POD 8080 暴露了 12 个端口。

3 --publish-all

部署环境为 Kubernetes 集群。集群中部署了 POD 8080 和 POD 8081。POD 8080 部署了 4 个容器，POD 8081 部署了 2 个容器。

4 --publish (-p)

HOST [][][][] [][][][][] [][][][]. [][] POD[][][][][][][][][].

2. PODMAN

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

publish rootful podman



podman network ls

port ☐ ☐ ☐ ☐ ☐

podman port -a

POD ☐☐☐☐ ☐☐ ☐☐☐☐

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

podman network reload

POD □□□□ □□□□ □□

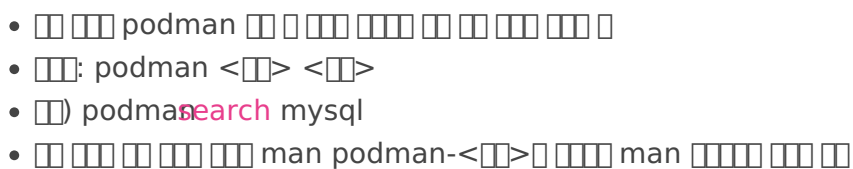
□□ □ □□□ □□□□ □□□□ □□□□□ podman □□.

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 列出所有 pod 的 ID。 pod 的 ID

podman start <POD 的 ID>

列出所有 pod。

podman inspect

inspect 列出 pod 的详细信息，如 IP 地址、端口等。

podman search

Registry(仓库)的搜索。 如 podman search docker.io/library/nginx:latest。 Internet 上的 external 仓库。

docker 仓库 library 的搜索。 docker.io/library/<仓库 ID> 的搜索。

列出所有 pod 的 ID

podman top <POD 的 ID>

列出所有 pod 的 ID

podman images

podman tag 列出所有 pod 的 ID

podman pull certbot/certbot:arm64v8-latest

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

列出所有 pod 的 ID

podman rmi <ID>

podman rmi 列出所有 pod 的 ID -f 强制删除所有 pod 的 ID。

POD(容器) 删除

容器删除命令为 `podman rm`。删除容器时，容器必须处于已停止状态。

`podman rm <POD容器 ID>`

删除容器时，如果容器处于运行状态，使用 `-f` 强制删除。

`-f` 选项与 `podman kill` 类似，`podman rm` 命令会先尝试杀死容器。

`podman rm -f <POD容器 ID>`

PODMAN 简介



Podman 是一个容器引擎

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 Pod。Pod 是在 HOST OS 上运行的容器。

restart

Podman 支持 restart 命令，用于重启已经运行的 Pod。

stop

Podman 支持 stop 命令，用于停止已经运行的 Pod。

kill

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