

Docker核心技术之数据管理



Docker核心技术之数据管理-课程概要

- Docker 数据卷简介
- Docker 数据卷管理
- Docker 数据卷注意事项
- 总结

Docker核心技术之数据管理

Docker 数据卷简介

为什么用数据卷

- 宿主机无法直接访问容器中的文件
- 容器中的文件没有持久化，导致容器删除后，文件数据也随之消失
- 容器之间也无法直接访问互相的文件

为解决这些问题，docker加入了数据卷(volumes)机制，能很好解决上面问题，以实现：

- 容器与主机之间、容器与容器之间共享文件
- 容器中数据的持久化
- 将容器中的数据备份、迁移、恢复等

数据卷的特点

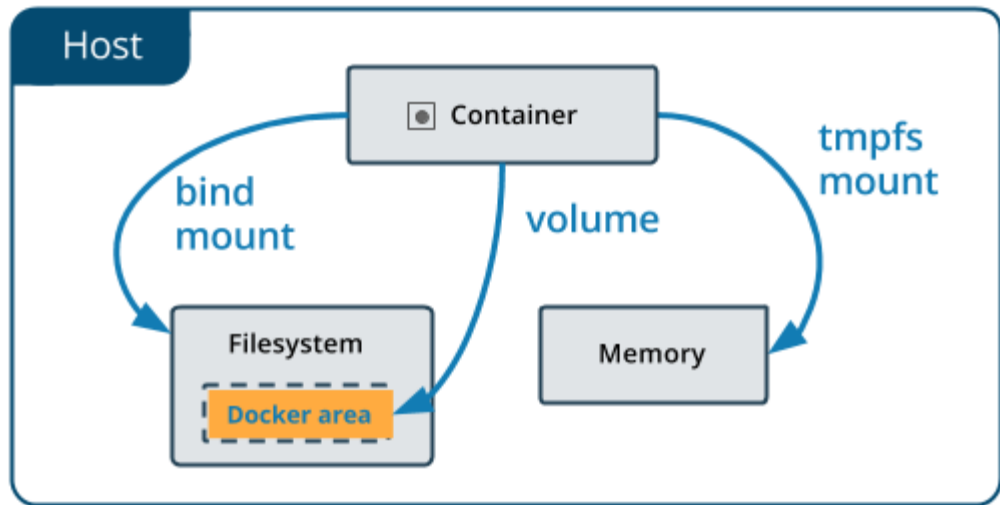
- 数据卷存在于宿主机的文件系统中，独立于容器，和容器的生命周期是分离的。
- 数据卷可以目录也可以是文件，容器可以利用数据卷与宿主机进行数据共享，实现了容器间的数据共享和交换。
- 容器启动初始化时，如果容器使用的镜像包含了数据，这些数据会拷贝到数据卷中。
- 容器对数据卷的修改是实时进行的。
- 数据卷的变化不会影响镜像的更新。数据卷是独立于联合文件系统，镜像是基于联合文件系统。镜像与数据卷之间不会有相互影响。

Docker核心技术之数据管理

Docker 数据卷管理

Docker挂载容器数据卷的三种方式

- **bind mounts**: 将宿主主机上的一个文件或目录被挂载到容器上。
- **volumes**: 由Docker创建和管理。使用 docker volume 命令管理
- **tmpfs mounts**: tmpfs 是一种基于内存的临时文件系统。tmpfs mounts 数据不会存储在磁盘上。



bind mounts方式挂载数据卷

- 利用docker run/create的参数为容器挂载数据卷

- 用法:

方式一: -v, --volume参数

-v 宿主机文件或文件夹路径:容器中的文件或者文件夹路径

方式二: --mount参数

--mount type=bind, src=宿主机文件或文件夹路径, dst=容器中的文件或者文件夹路

径

注意: src指定的文件和路径必须提前创建或存在

- 命令演示:

```
[root@centos-linux ~]# docker run --rm -dti -v /root/host_dir:/root/c_dir centos bash
9944078a259d016b2f9e417de284065aac8b3cd6ef95ffb17324db4171ba383f
[root@centos-linux ~]# mkdir /root/host_dir2
[root@centos-linux ~]# docker run --rm -dti --mount type=bind,src=/root/host_dir2,dst=/root/c_dir2 centos bash
427a1c2eaf8074bd6e65dfbd57b5c56b56dbe9dd508a093362740c77a5f2d056
[root@centos-linux ~]#
```


volumes方式挂载数据卷

- 利用docker run/create为容器挂载数据卷
- 用法：

方式一： -v, --volume 参数

-v VOLUME-NAME:容器中的文件或者文件夹路径

方式二： --mount 参数

--mount type=volume, src=VOLUME-NAME, dst=容器中的文件或者文件夹路径

- volume对象管理：

docker volume 命令管理volume数据卷对象

docker volume **create**

创建数据卷对象

docker volume **inspect**

查看数据卷详细信息

docker volume **ls**

查看已创建的数据卷对象

docker volume **prune**

删除未被使用的数据卷对象

docker volume **rm**

删除一个或多个数据卷对象

tmpfs mount方式挂载数据卷

- 利用docker run/create为容器挂载数据卷
- 用法：

--mount type=tmpfs, dst=PATH

共享其他容器的数据卷-数据卷容器

- 利用docker run/create 的--volumes-from参数指定数据卷容器
- 用法：

docker run/create --volumes-from CONTAINER

Docker核心技术之数据管理

Docker 数据卷注意事项

数据卷使用注意

Docker的数据卷更多会是使用volumes方式来进行使用。使用时需注意：

- 如果挂载一个空的数据卷到容器中的一个非空目录中，那么这个目录下的文件会被复制到数据卷中。
- 如果挂载一个非空的数据卷到容器中的一个目录中，那么容器中的目录中会显示数据卷中的数据。如果原来容器中的目录中有数据，那么这些原始数据会被隐藏掉。

这两个规则都非常重要，灵活利用第一个规则可以帮助我们初始化数据卷中的内容。掌握第二个规则可以保证挂载数据卷后的数据总是你期望的结果。

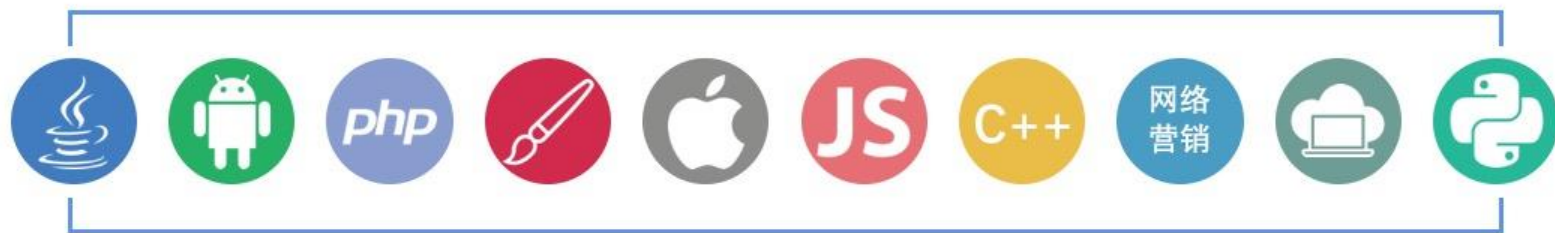
Docker核心技术之数据管理

总结

Docker概览-总结

重点掌握

- 数据卷特征和简介
- mount方式绑定数据卷
- 数据卷挂载的三种形式
- **数据卷使用的注意事项**



Thank you!

改变中国IT教育，我们正在行动

BOXUEGU.COM