Docker数据卷

教学内容

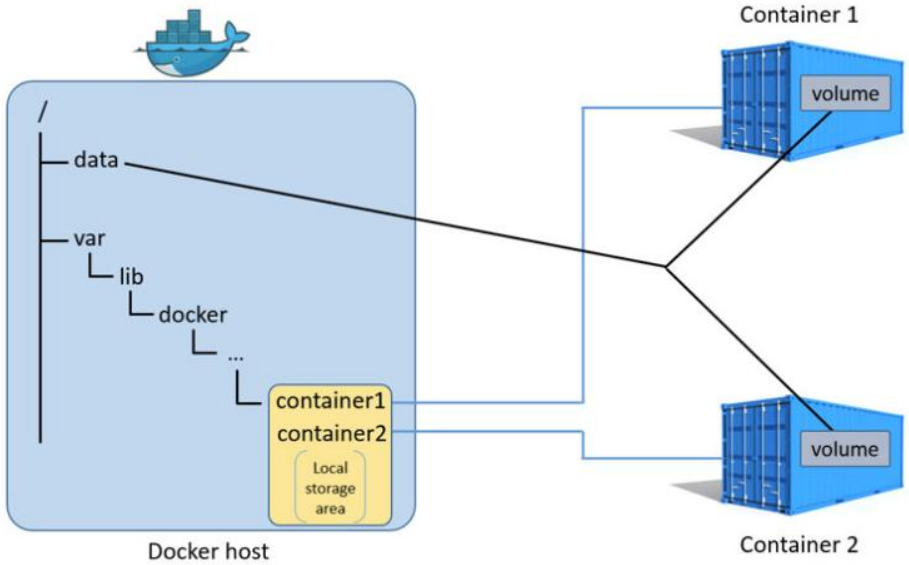
- 第一节 数据卷基本概念
- 第二节 数据卷操作指令
- 第三节 数据卷应用案例

容器中的数据问题

- 一个容器运行了一段时间，肯定会产生一些数据，比如日志、数据库数据、新改的配置文件等等，如果这些数据文件存放在容器中，当我们删除容器时，这些数据也会被随之删除。
- 在docker中，提供了一种存储数据的方法，叫做“数据卷”，可以达到数据共享的目的。

数据卷基本概念

我们可以把“数据卷”Data Volumes理解成“宿主机中的目录”，当把某个卷和容器中的某个目录建立映射关系后，就相当于把宿主机中的某个目录和容器中的某个目录建立了映射关系



数据卷基本概念

数据卷提供了很多有用的特性：

- 数据卷可以在容器之间共享和重用，容器间传递数据将变得高效与方便；
- 对数据卷内数据的修改会立马生效，无论是容器内操作还是本地操作；
- 对数据卷的更新不会影响镜像，解耦开应用和数据；
- 卷会一直存在，直到没有容器使用，可以安全地卸载它。

数据卷操作命令

数据卷操作的基本语法为： docker volume [COMMAND]，其中COMMAND可选值：

- create：创建一个volume；
- inspect：显示一个或多个volume的信息
- ls：列出所有的volume
- prune：删除未使用的volume
- rm：删除一个或多个指定的volume

数据卷操作示例

- 创建数据卷：docker volume create testA；
- 查看数据卷：docker volume ls
- 查看对应卷的详细信息：docker volume inspect testA
- 在linux的docker主机中创建一个卷时，其在宿主机对应的目录（挂载点）路径为/var/lib/docker/volumes/卷名/_data

挂载卷

- 在创建容器时，可以通过--volume或 -v 参数挂载一个数据卷到某个容器目录
- docker run --name testAcon -v testA:/data -d redis
- 上述命令表示创建一个名为testAcon的容器，将testA卷映射到testAcon容器的/data目录中
- 注意：如果卷映射的目录在容器中不存在时，会自动在容器中创建对应的目录
- 一个容器可以使用多个卷，只需要多次使用-v选项指定即可
- docker run --name testBcon -v testA:/data -v testB:/var/log -d redis
- 当指定的卷不存在时，docker会自动创建对应的卷，上述命令中的testB数据卷会被自动创建

绑定挂载

- 前面创建的数据卷都存放在/var/lib/docker/volumes目录中，这个目录是固定的，它们都能被docker volume命令管理。
- docker还有一种映射宿主机目录的方法，这种方法被称之为”绑定挂载”,绑定挂载能够将指定的宿主机目录挂载到容器中,只需要将卷名替换成宿主机上的目录路径即可
- docker run -d --name testAcon -v /root/test1:/data1 redis
- 上述命令将宿主机的/root/test1目录映射到容器的/data1目录中
- 绑定挂载不会生成任何卷，它直接将指定的宿主机目录映射到容器中，所以，docker volume命令无法查看或管理到绑定挂载的路径

绑定挂载

- 官方建议使用卷，而不是绑定挂载，但是，绑定挂载有一个优势，就是绑定挂载可以直接将宿主机中的文件（非目录）直接挂载到容器中，比如，将宿主机中的/etc/localtime文件映射到容器中的/etc/localtime文件
- docker run -d --name testAcon -v /etc/localtime:/etc/localtime alpine
- 通常，使用绑定挂载就是为了将宿主机中的配置文件挂载到容器中，如果是整个目录的数据，建议使用卷，卷只能映射目录，不能映射文件。

nginx示例

- 创建容器并挂载数据卷到容器内的HTML目录
- docker run --name myng -v html:/usr/share/nginx/html -p 80:80 -d nginx
- 进入html数据卷所在位置，并修改HTML内容
- cd /var/lib/docker/volumes/html/_data

mysql示例

- https://hub.docker.com/_/mysql