5 容器数据卷

## 1 什么是容器数据卷

docker理念回顾：

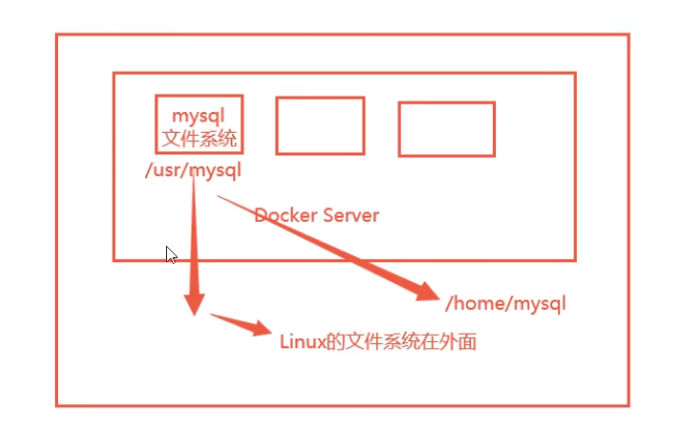
将应用和环境打包一个镜像，直接运行使用。

数据？如果数据在容器中，那么当删除容器后，数据会丢失！

需求：数据可以持久化 如:将mysql数据可以存储在本地

容器之间可以有一个数据共享技术，将docker容器中产生的数据，同步到本地。

这就是卷技术！说白了就是目录的挂载，将我们容器内的数据，挂在到宿主机上，



**总结：容器和持久化和同步操作！容器间也可以使数据共享的！**

## 2 使用数据卷

### 方式一 命令挂载 -v

直接使用命令来挂载 使用-v参数

好处：我们以后修改只需要在本地修改即可，容器内会自动同步

## 3 实战安装MySQL

思考：MySQL的数据持久化问题

通过案例发现，我们挂载到本地的数据卷依旧没有丢失，这就实现了容器数据持久化功能

## 4 具名和匿名挂载

所有的docker容器数据卷，在没有指定宿主机目录的情况下，linux 本地数据默认存储在/var/lib/docker/volumes/xx/\_data位置

我们通过具名挂载可以方便的找到本地数据卷，所以大多情况下都使用具名挂载

## 5 初识别dockerfile

之前使用docker commit 来构建了一个镜像，现在通过dockerfile构建

dockerfile来构建docker镜像的构建文件！写一个命令脚本，run！

### 方法二 dockerfile挂载数据卷

通过这个脚本，可以生成一个镜像！

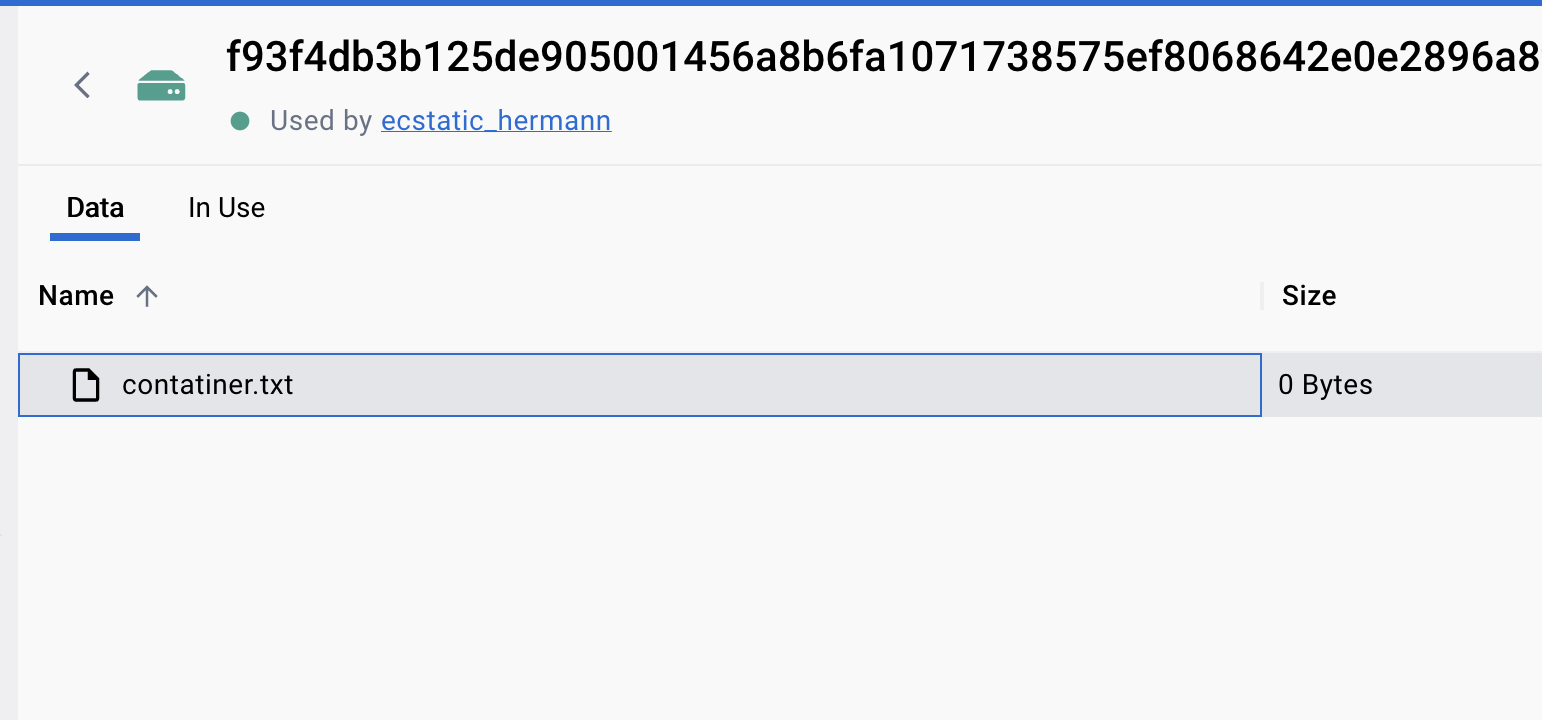
这个数据卷和外部一定有一个同步目录！

脚本里：

VOLUME ["volume01","volume02"]

这一行是只给定了容器内路径，那么就是匿名挂载

测试一下刚才的文件同步到本地!



这种方法未来使用的非常多，因为我们通常会构建自己的镜像！

假设构建镜像时没有挂载数据卷，要手动镜像挂载 -v 卷名:容器内路径！

## 6 数据卷容器

### 同步数据



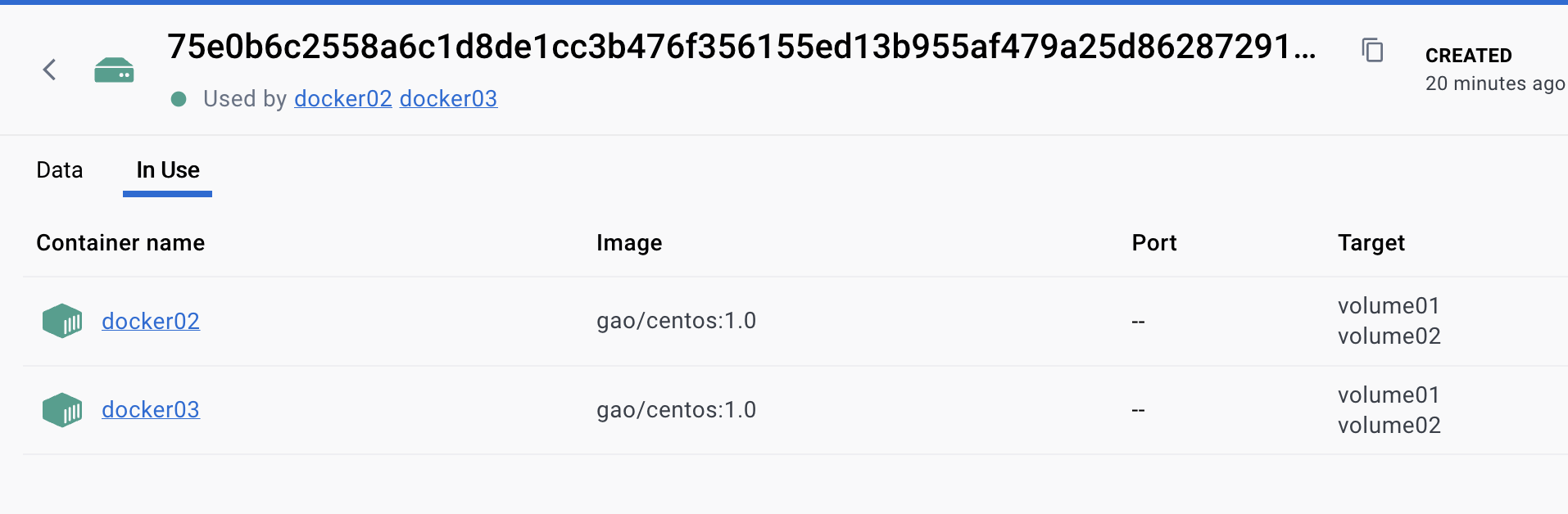
--volumes-from：子类容器继承父类容器,这也就叫 **数据卷容器**

通过--volumes-from 就可以实现容器间的数据共享

删掉docker01容器，docker02、docker03的文件依旧存在

因为通过docker inspect 容器id我们发现，他们容器的数据卷都挂载到同一个宿主机地址

删除了（docker01）：



### 多个mysql实现数据共享

结论：

容器之间配置信息的传递，数据卷容器的声明周期一直持续好没有容器为止！

但是，一旦持久化到本地，这个时候容器内的数据不会删除，这就是容器数据卷的优势。