



Docker 数据管理

Docker 数据管理

生产环境中使用Docker的过程，往往需要对数据进行持久化，或者需要在多个容器之间进行数据共享，这必然涉及容器的数据管理操作。

这一章介绍如何在 Docker 内部以及容器之间管理数据，在容器中管理数据主要有两种方式：

数据卷（Data volumes）：容器内数据之间映射到本地主机环境；

数据卷容器（Data volume containers）使用特定容器维护数据卷。

Docker 数据管理

数据卷

数据卷是一个可供一个或多个容器使用的特殊目录，它绕过 UFS，可以提供很多有用的特性：

数据卷可以在容器之间共享和重用

对数据卷的修改会立马生效

对数据卷的更新，不会影响镜像

数据卷默认会一直存在，即使容器被删除

*注意：数据卷的使用，类似于 Linux 下对目录或文件进行 mount，镜像中的被指定为挂载点的目录中的文件会隐藏掉，能显示看的是挂载的数据卷。

Docker 数据管理

数据卷

运行一个容器，并创建一个数据卷挂载到容器的某个目录上

```
#docker run -dti -v /web centos:7.0  
/bin/bash
```

利用 centos:7.0 的镜像运行一个容器，并在容器内创建一个数据卷挂载到容器的 /web 目录上

Docker 数据管理

1. 挂载本地的目录到容器里

```
#docker run -itd -v /data:/data/ 镜像名  
/IMAGE_ID bash
```

可以用其他容器再挂载这个容器的目录
实现多个容器和本地目录同步、共享 如
挂载其它容器的目录

-d 选项表示后台运行

-v 选项用来指定挂载(创建映射关系)的目录

Docker 数据管理

- `/data:/data/` 宿主机目录:容器目录, :前面的`/data/`为本地目录:后面的`/data/`为容器里的目录, 名字不一定要相同 只要自己指定好对应的目录就可以 宿主机要手动去创建这个目录 容器会自动生成这个目录

镜像名/`IMAGE_ID` 为 `#docker images` 显示的 `REPOSITORY` 列的 镜像名 或者 `IMAGE_ID` 列的 `id`

Docker 数据管理

```
[root@localhost ~]# docker run -tid -v /data:/data1/ centos bash
aa6ad30230f0eaaf47f35e26a97928bea620bcd5ad0c8c1a3f97968076e77e6
[root@localhost ~]# touch /data/1.txt
[root@localhost ~]# echo "abcdf" > /data/1.txt
[root@localhost ~]# docker attach aa6ad30230f bash
[root@aa6ad30230f0 /]# ls /data1/
1.txt mysql
[root@aa6ad30230f0 /]# cat 1.txt
cat: 1.txt: No such file or directory
[root@aa6ad30230f0 /]# cat /data1/1.txt
abcdf
```

Docker 数据管理

```
[root@aa6ad30230f0 /]# touch /data/2.txt
```

```
[root@aa6ad30230f0 /]# exit
```

```
exit
```

```
[root@localhost ~]# ls /data/
```

```
1.txt 2.txt mysql
```

```
[root@localhost ~]# docker stop aa6ad30230f
```

```
aa6ad30230f
```

```
[root@localhost ~]# ls /data
```

```
1.txt 2.txt mysql
```

```
[root@localhost ~]# docker rm aa6ad30230f
```

```
aa6ad30230f
```

```
[root@localhost ~]# ls /data/
```

```
1.txt 2.txt mysql
```


Docker 数据管理

删除数据卷

数据卷是被设计用来持久化数据的，它的生命周期独立于容器，Docker不会在容器被删除后自动删除数据卷，并且也不存在垃圾回收这样的机制来处理没有任何容器引用的数据卷。如果需要在删除容器的同时移除数据卷。可以在删除容器的时候使用 `docker rm -v` 这个命令。无主的数据卷可能会占据很多空间，要清理会很麻烦。

Docker 数据管理

删除数据卷

如果删除了挂载的容器，数据卷并不会被自动删除。如果要删除一个数据卷，需要使用

```
$ docker rm -v
```

Docker 数据管理

Data Volume Containers --- 数据卷容器

如果用户需要在容器之间共享一些持续更新的数据，最简单的方法就是使用数据卷容器，其实数据卷容器就是一个普通的容器，只不过是专门用它提供数据卷供其他容器挂载使用

Docker 数据管理

创建一个名为 dbdata 的数据卷，并在其中创建一个数据卷挂载到 /dbdata 上

```
#docker run -dti -v /dbdata --name dbser centos:7.0
```

```
*****#ls
```

进入到 dbdata 容器中去查看数据卷是否创建成功

--name NAME 给容器指定一个有特点方便识别记忆的名字

Docker 数据管理

Data Volume Containers --- 数据卷容器

然后在其他容器中使用 `--volume-from` 去挂载 `dbdata` 容器中的 `/dbdata` 数据卷

创建 `db1` & `db2` 两个容器，并挂载 `/dbdata` 数据卷到本地

```
#doocker run -dti --volumes-from dbser --name db1 centos:7.0
```

```
#doocker run -dti --volumes-from dbser --name db2 centos:7.0
```

此时，容器 `db1` 和 `db2` 同时挂载了同一个数据卷到本地相同 `/dbdata` 目录。三个容器任何一个目录下的写入，都可以时时同步到另外两个

Docker 数据管理

Data Volume Containers --- 数据卷容器
如果删除了某个挂载了的容器（dbdata，db1 和 db2），数据卷不会被自动删除
如想删除用 `docker rm -v`

Docker 数据管理

利用数据卷容器迁移数据 --- 数据的备份 / 恢复，以实现数据的迁移

1. 备份数据卷

```
#docker run -dti -v /dbdata --name dbser  
centos:7.0 /bin/bash
```

创建一个数据卷容器，指定数据卷为 /dbdata，并向 /dbdata 中写入数据以备我们后续需要备份使用

```
#docker run -dti --volume-from dbser -v  
$(pwd):/backup --name  
beifen Centos:7.0 /bin/bash
```

Docker 数据管理

首先利用 centos:7.0 镜像生成一个名为 beifen 的容器，并挂载数据卷

容器 dbser 的数据卷和将容器内的 /backup 目录映射到自己当前所处的目录（\$(pwd)）

```
****#tar cvf /backup/backup.tar /dbdata
```

进入容器，将容器的 /dbdata 目录打包到 /backup 目录取名 backup.tar

Docker 数据管理

恢复

```
# docker run -dti -v /dbdata --name dbload  
centos:7.0 /bin/bash
```

利用 centos:7.0 镜像生成一个名为 dbload 的容器，并在本地建立一个数据卷 /dbdata

```
# docker run -dti --volume-from dbload -v  
$(pwd):/backup centos:7.0  
# tar xvf /backup/backup.tar
```

Doccker 数据管理

利用 centos:7.0 镜像生成一个容器，挂载 dbload 的数据卷，并将当前所在目录挂载给容器的 /backup 目录，然后解压 /backup 目录里的 backup.tar 文件