



Docker Volume入门用法详解

 *devin0910* · 2018-09-14 11:24:10

[docker](#) [docker-volume](#)

Docker Volume能让容器从宿主主机中读取文件或持久化数据到宿主主机内，让容器与容器产生的数据分离开来。你可以简单地把它理解为linux服务器上的挂载点。一个容器可以挂载多个不同的目录。Volume的生命周期是独立于容器的生命周期之外的，即使容器删除了，volume也会被保留下来，Docker也不会因为这个volume没有被容器使用而回收。在容器中，添加或修改这个文件夹里的文件也不会影响容器的联合文件系统。

数据卷不使用分层文件系统，这对经常读取和写入的数据很有利。在开发过程中，你可以把代码目录挂载到容器中，这样，如果你更改代码，容器会实时地得到文件修改的反馈。容器中的挂载点必须是绝对路径，不支持相对路径。宿主机上的地址可以是一个绝对路径，也可以是一个数据卷名(如果数据卷不存在，docker会自动创建数据卷)。

注意：不要在Dockerfile中指定挂载一个host主机的目录。不够灵活，因为在别的主机上不一定有这样一个目录。

创建数据卷

在web开发过程中，通常需要一个web服务器，下面我们使用nginx镜像来做示例，把web开发目录挂载到nginx容器中。

```
docker run --rm --name webserver -p 80:80 -v /var/www:/usr/share/nginx/html:ro -d nginx
```

前面的/var/www为宿主机上的绝对路径，/usr/share/nginx/html为容器上的绝对路径，ro指定只读挂载。在本机上创建一个index.html,然后就可以直接通过浏览器访问了，修改index.html文件，刷新浏览器，可以马上看到文件的更新。

docker volume命令有所有关于volume的命令。

```
docker volume --help
```

Commands:	
create	创建一个数据卷
inspect	打印一个或多个数据卷的详细信息
ls	列出所有数据卷
prune	删除所有未使用的数据卷
rm	删除一个或多个数据卷

如何创建一个数据卷？

```
docker volume create www-data
```

列出所有数据卷

```
docker volume ls
```

DRIVER	VOLUME NAME
local	www-data

查看数据卷在宿主机上的路径：



```
docker inspect www-data

[
  {
    "Driver": "local",
    "Labels": {},
    "Mountpoint": "/var/lib/docker/volumes/www-data/_data",
    "Name": "www-data",
    "Options": {},
    "Scope": "local"
  }
]
```

注：每创建一个volume，docker会在/var/lib/docker/volumes/下创建一个子目录，默认情况下目录名是一串UUID。如果指定了名称，则目录名是volume名称（例如上面的www-data）。volume里的数据都存储在这个子目录的_data目录下。

MountPoint 宿主机上的路径为/var/lib/docker/volumes/www-data/_data，数据卷的名称为www-data。

把这个数据卷挂载到一个新的nginx容器中

```
docker run --rm --name web -p 80:80 -v www-data:/usr/share/nginx/html:ro -d nginx
```

查看数据卷对应的宿主机上的目录

```
ls -al /var/lib/docker/volumes/www-data/_data

-rw-r--r-- 1 root root  537 Jul 11 21:06 50x.html
-rw-r--r-- 1 root root  612 Jul 11 21:06 index.html
```

修改其中的index.html然后刷新浏览器，就可以看到我们刚才的更改。

完成上面一系列的操作之后，在命令行输入`docker rm web`删除容器，再次查看上面的路径，发现文件还是会保留在那儿，并不会因为容器删除了数据卷也随之删除。可以通过`-v`选项，在删除的时候也删除关联的数据卷（注意：如果是命名的数据卷，则不会被删除，上面的示例中，即使你在删除容器的时候指定了`-v`选项，数据卷`www-data`也不会被删除）。下面简单的讲解一个能通过`-v`删除的示例：

```
docker run --name web1 -v /usr/share/nginx/html -d nginx

// 查看数据卷在宿主机上的路径：
docker inspect -f {{.Mounts}} web1

[{"volume": "853f05868bc1ab4de84a9e646ce022183a2fcbe4e0edce629ff8e0a8f33fdf72",
  "source": "/var/lib/docker/volumes/853f05868bc1ab4de84a9e646ce022183a2fcbe4e0edce629ff8e0a8f33fdf72/_data",
  "destination": "/usr/share/nginx/html",
  "mode": "local",
  "rw": true}]
```

然后通过`docker rm -fv web1`删除容器，`-f`可以删除正在运行的容器。现在宿主机上已找不到上面打印出来的目录了。

```
// 列出所有不被引用的数据卷
docker volume ls -f dangling=true

DRIVER              VOLUME NAME
local               www-data
```

删除所有不被使用的数据卷

```
docker volume prune
```

再通过`docker volume ls`发现所有的数据卷已经被删除了。

```
// 删除所有的数据卷
docker volume rm $(docker volume ls -q)

// 停止所有的容器
docker stop $(docker ps -a -q)

// 删除所有的容器
docker rm $(docker ps -a -q)
```



-q选项只打印容器id。

数据容器

数据容器是一类特殊的容器，它并不运行任何程序，只是用来保存数据，可以通过它让不同的容器可以共享数据。

创建一个数据容器：

```
docker create -v /dbdata --name dbstore busybox
```

通过--volumes-from 来挂载数据容器

```
docker run -it --volumes-from dbstore ubuntu /bin/bash
```

挂载完成后，你可以在容器中看到/dbdata目录，这里面的文件修改会保存回dbstore容器中。

备份与还原数据

运维的日常工作中通常少不了数据的备份，还原，迁移等操作。下面介绍如何在数据卷中进行上面的操作：

```
docker run --rm --volumes-from dbstore -v $(pwd):/backup ubuntu tar cvf /backup/backup.tar /dbdata
```

我们通过--volumes-from挂载要备份的数据卷，-v挂载一个宿主机上要保存备份文件的路径，然后在容器里运行tar cvf /backup/backup.tar /dbdata保存/dbdata中的数据到/backup目录中，而/backup正好是我们前面挂载的宿主主机上的路径，这样就把数据卷备份下来了。--rm选项表示备份完数据之后删除这个容器。

下面我们来介绍如何把备份数据还原到一个新的容器中： 首先我们启动一个新的容器：

```
docker run -v /dbdata --name dbstore2 ubuntu /bin/bash
```

然后把备份数据还原到新的容器中

```
docker run --rm --volumes-from dbstore2 -v $(pwd):/backup ubuntu bash -c "cd /dbdata && tar xvf /backup/backup.tar --strip 1"
```

<http://www.tricksofthetrades.net/2016/03/14/docker-data-volumes/>

你想说点什么



评论

码客教程

高品质的程序员教程社区。我们提供优质的教程致于开发者学习、分享、探讨的一个全新平台。

网站相关

联系我们

友情链接

- SegmentFault
- StackOverflow
- Github
- Oschina
- LeetCode
- Android Blog
- Hacker News
- Bootstrap
- sci
- 羊分享
- 雨笙Blog



