



6-1、docker容器---->数据挂载---->宿主机

- <https://baijiahao.baidu.com/s?id=1642180660416309492&wfr=spider&for=pc>

1、目的

- 将容器内的文件挂载到宿主机上，达到文件内容共享
- 将容器内的目录挂载到宿主机上，达到目录中文件的共享目的
- 双向绑定实现数据共享，一方写另一方共享

2、如何挂载

- docker容器启动时，如果要将容器内的数据挂载宿主机上，可以通过-v参数指定

3、挂载方式

- 如何确定是那种挂载方式

┌ -v 容器内绝对路径 #匿名挂载

-v 卷名:容器内绝对路径 #具名挂载

-v 宿主机绝对路径:容器内绝对路径 #指定路径挂载

--mount type=volume/bind, src=, dst=

使用--mount type=volume,dst=/soft/ 替换 -v /soft (替换匿名挂载)

使用--mount type=volume,src=hgm-test,dst=/soft/ 替换 -v hgm-test:/soft (替换具名挂载)

使用--mount type=bind,src=/hgm-test,dst=/soft/ 替换 -v /hgm-test:/soft (替换指定挂载)

如果使用--mount时type和src,dst不对应会报错



+

- 匿名挂载
 - -v只指定一个绝对路径时
 - docker run -it -v /hgm nginx:1.0 /bin/bash



- 容器内路径：/hgm，如果没有会自动创建
- 挂载到宿主机的路径：/var/lib/docker/volumes/**xxxx**
 - /var/lib/docker/volumes下是所有挂载的数据卷目录，比如**xxxx**目录，**卷目录名是自动生成的随机字符串**，每个卷目录下的_data目录存储的是挂载的数据
- ∴ ○ 通过docker volume ls，查看宿主机上所有被挂载的数据卷
- 通过docker volume rm 卷名，删除挂载的数据卷
- 通过docker volume inspect 卷名，查看主机上挂载的数据卷信息
- 通过docker inspect **xxxx**，查看宿主机上挂载的数据卷信息，(源)宿主机上挂载的数据卷的位置/var/lib/docker/volumes/xxxx，(目的地)容器内的位置/hgm
- 具名挂载（指定的卷名）
 - -v 指定第一个是相对路径，第二个是绝对路径
 - 其实它不是一个相对路径，而是随机生成的一个目录名（**hgm-test**）作为数据卷名
 - docker run -it -v **hgm-test:/soft** nginx:1.0 /bin/bash
 - 容器内路径：/soft，如果没有会自动创建
 - 挂载到宿主机的路径：/var/lib/docker/volumes/hgm
 - /var/lib/docker/volumes下是所有挂载的数据卷目录，此时卷目录名是指定的**hgm-test**，此目录下的_data目录存储的是挂载的数据
 - 通过docker volume ls，查看宿主机上所有被挂载的数据卷
 - 通过docker inspect **hgm-test**，查看宿主机上挂载的数据卷hgm-test的信息，(源)宿主机上挂载的数据卷的位置/var/lib/docker/volumes/**hgm-test**目录，(目的地)容器内的位置/soft
- 指定挂载



- -v指定两个绝对路径时
- 前面是挂载到宿主机的路径，后面是容器内的路径，如果路径没有会自动创建
- `docker run -it -v /hgm:/soft nginx:1.0 /bin/bash`
- 挂载到宿主机的位置：`/hgm`

- 挂载失败的方式

- -v指定第一个是绝对路径，第二个是相对路径
- `docker run -it -v /hgm:soft nginx:1.0 /bin/bash`
- 如果容器内目录hgm为相对路径会报错，会提示hgm不是一个绝对路径，所谓的绝对路径，必须以下斜线“/”开头。

4、三种挂载方式，容器启动时数据同步的方向：

- 匿名和具名--->容器到宿主机
- 指定挂载名--->宿主机到容器
- 匿名挂载和具名挂载由于宿主机的被挂载位置是docker自动指定的`/var/lib/docker/volumes/xxx/_data`，所以容器启动挂载时，以容器中的挂载目录下的数据为主，将数据同步覆盖到宿主机的被挂载目录_data下，后续不管操作哪一方的目录才是双向绑定，数据共享。
- 指定挂载，是手动指定好了宿主机被挂载的位置，所以容器启动挂载时，以宿主机中的挂载目录为主，将目录下的数据同步覆盖到容器的挂载目录位置，后续不管操作哪一方的目录才是双向绑定，数据共享。

5、挂载时，只有具名挂载和指定挂载的格式符合设置容器的权限，设置容器内能否修改挂载目录。

- `docker run -it -v hgm:/soft:ro nginx:1.0 /bin/bash` //格式正确，最优，见4讲解
- `docker run -it -v /hgm:ro nginx:1.0 /bin/bash` //格式错误，因为会认为ro是容器内的挂载目录，但只能是绝对路径
- `docker run -it -v /hgm:/soft:ro nginx:1.0 /bin/bash` //格式正确，不是最优，见4讲解

4、设置容器权限，挂载出来的内容有了限定，只能单向操作，只能宿主机操作



```
ro readonly #只读  
rw readwrite #可读可写
```

- 默认挂载的路径权限为读写，即容器内或宿主机都可以写操作，一方写另一方数据共享。
- 如果指定为只读可以用：ro，这里示例用的是指定挂载的方式
- `docker run -it -v /hgm:/soft:ro nginx:1.0 /bin/bash`
- 只能宿主机在/hgm写操作，容器内是无法在/soft写操作的，只有只读的权限

5、容器停止或销毁，宿主机上的挂载点目录依然存在

6、重启docker服务，宿主机上的挂载点目录依然存在