9.数据卷和数据卷容器

1.容器中管理数据主要有两种方式：

- 数据卷(Data volumes)
- 数据卷容器（Data volume containers)

2.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：

- a.数据卷可以在容器之间共享和重用
- b.对数据卷的修改会立马生效
- c.对数据卷的更新，不会影响镜像
- d.卷会一直存在，直到没有容器使用

*数据卷的使用，类似于 Linux 下对目录或文件进行 mount。

3.创建数据卷

创建的数据卷默认是存在在容器中

```
#-v标记创建一个数据卷并且挂载到容器里。
#--name指定容器的名称
#-d是指守护状态下运行（daemon）
#启动之后，使用docker attach命令进入守护状态运行的容器中
docker run -it -P -d --name data1 -v /datas my/centos:v1.1 /bin/bash
```

```
[root@bigdata-4 ~]# docker run -it -P -d -v /datas my/centos:v1.1 /bin/bash
4d3c3cdfc757b102689f1262a5a3171df4de53bf5a8e192a1d3455c83f67876b
[root@bigdata-4 ~]# docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS
MES
4d3c3cdfc757        my/centos:v1.1      "/bin/bash"              15 seconds ago      Up 14 seconds
stracted_pare
b3af7de1751c        registry            "/entrypoint.sh /e..."   2 hours ago         Up 2 hours          0.0.0.0:5000->5000/tc
rene_bohr
[root@bigdata-4 ~]# docker attach 4d3c3cdfc757b102689f1262a5a3171df4de53bf5a8e192a1d3455c83f67876b
[root@4d3c3cdfc757 /]# ls
anaconda-post.log  bin  datas  dev  etc  home  lib  lib64  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
[root@4d3c3cdfc757 /]# cd /datas
[root@4d3c3cdfc757 datas]# ll
total 0
[root@4d3c3cdfc757 datas]# cat >1.txt
aaaaaaa
^C
[root@4d3c3cdfc757 datas]# ls
1.txt
[root@4d3c3cdfc757 datas]# cat 1.txt
aaaaaaa
[root@4d3c3cdfc757 datas]#
```

```
[root@bigdata-4 ~]# docker run -it -P -d --name data1 -v /datas my/centos:v1.1 /bin/bash
230c6f859ac5b311ddc394b8fc5083331438e0cc36543306c9cd8681188c0015
[root@bigdata-4 ~]# docker attach data1
[root@230c6f859ac5 /]# ls
anaconda-post.log  bin  datas  dev  etc  home  lib  lib64  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
[root@230c6f859ac5 /]# cd /datas
[root@230c6f859ac5 datas]# ls
[root@230c6f859ac5 datas]# cat > 1.txt
aaaaa
^C
[root@230c6f859ac5 datas]# cat 1.txt
aaaaa
[root@230c6f859ac5 datas]#
```

除此之外，还可以挂载主机目录导容器中作为数据卷

```
docker run -d -P -it --name web2 -v /root/docker:/opt/docker my/centos:v1.1 /bin/bash
```

docker挂载的数据卷默认是可读可写的

```
[root@bigdata-4 ~]# docker run -d -P -it --name web2 -v /root/docker:/opt/docker my/centos:v1.1 /bin/bash
7372b0146c6243525d8d84bb4727e345026c0ddc3d27a87004282cea0eac0a5e
[root@bigdata-4 ~]# docker attach web2
[root@7372b0146c62 /]# ls
anaconda-post.log  bin  dev  etc  home  lib  lib64  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
[root@7372b0146c62 /]# cd /opt/docker/
[root@7372b0146c62 docker]# ls
registry
[root@7372b0146c62 docker]# cd registry/
[root@7372b0146c62 registry]# ll
total 0
[root@7372b0146c62 registry]# vi 1.txt
[root@7372b0146c62 registry]# cat 1.txt
jfdludlajfdljfdlajf;d
jfdlkjfladjfld
[root@7372b0146c62 registry]#
```

映射目录，数据同步

```
drwxr-xr-x 2 root root 6 Aug 13 11:35 registry
[root@bigdata-4 docker]# cd registry/
[root@bigdata-4 registry]# ll
total 0
[root@bigdata-4 registry]# ls
1.txt
[root@bigdata-4 registry]# pwd
/root/docker/registry
[root@bigdata-4 registry]# cat 1.txt
jfdludlajfdljfdlajf;d
jfdlkjfladjfld
[root@bigdata-4 registry]#
```

还可以维挂载主机目录容器中数据卷指定权限,若要指定为只读，可以加权限限制：ro

```
docker run -d -P -it --name web3 -v /root/docker:/opt/docker:ro my/centos:v1.1
/bin/bash
```

```
[root@bigdata-4 ~]# docker run -d -P -it --name web3 -v /root/docker:/opt/docker:ro my/centos:v1.1 /bin/bash
808ead49de83353994a71e9d83cefb0e1dc0fc0a313139049cbc5fc890f1293b
[root@bigdata-4 ~]# docker attach web3
[root@808ead49de83 /]# cd /opt/docker/
[root@808ead49de83 docker]# vi 1.txt
[root@808ead49de83 docker]# cat >1.txt
bash: 1.txt: Read-only file system
[root@808ead49de83 docker]#
```

也可以挂载单个文件到容器

```
docker run -d -P -it --name web4 -v /root/.bash_history:/data:ro my/centos:v1.1
/bin/bash
```

```
exit
[root@bigdata-4 ~]# docker run -d -P -it --name web4 -v /root/.bash_history:/data:ro my/centos:v1.1 /bin/bash
fd3c837d8ccae37064a2802cb1ffff31f94c75877e972a4e522e1b5967446508
[root@bigdata-4 ~]# docker attach web4
[root@fd3c837d8cca /]# ls
anaconda-post.log  bin  data  dev  etc  home  lib  lib64  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
[root@fd3c837d8cca /]# ll
total 16
-rw-r--r--    1 root root 12082 Mar  5 17:36 anaconda-post.log
lrwxrwxrwx    1 root root     7 Mar  5 17:34 bin -> usr/bin
-rw-------.   1 root root  3956 Aug 13 02:18 data
drwxr-xr-x    5 root root   360 Aug 13 07:46 dev
drwxr-xr-x    1 root root    66 Aug 13 07:46 etc
drwxr-xr-x    2 root root     6 Apr 11  2018 home
lrwxrwxrwx    1 root root     7 Mar  5 17:34 lib -> usr/lib
lrwxrwxrwx    1 root root     9 Mar  5 17:34 lib64 -> usr/lib64
drwxr-xr-x    2 root root     6 Apr 11  2018 media
drwxr-xr-x    2 root root     6 Apr 11  2018 mnt
drwxr-xr-x    2 root root     6 Apr 11  2018 opt
dr-xr-xr-x  124 root root     0 Aug 13 07:46 proc
dr-xr-x---    3 root root   147 Aug 12 04:08 root
drwxr-xr-x   12 root root   163 Aug 12 04:02 run
lrwxrwxrwx    1 root root     8 Mar  5 17:34 sbin -> usr/sbin
drwxr-xr-x    2 root root     6 Apr 11  2018 srv
dr-xr-xr-x   13 root root     0 Aug 13 03:36 sys
drwxrwxrwt    7 root root   132 Aug 12 04:02 tmp
drwxr-xr-x   13 root root   155 Mar  5 17:34 usr
drwxr-xr-x   18 root root   238 Mar  5 17:34 var
[root@fd3c837d8cca /]# cd data
bash: cd: data: Not a directory
[root@fd3c837d8cca /]# cat data
vi /etc/hostname
vi /etc/sysconfig/network-scripts/ifcfg-ens33
reboot
jps
```

4.数据卷容器：是一个正常的容器，专门用来提供数据卷供其他容器挂载 建立数据卷容器：

```
docker run -it -d -v /database --name db my/centos:v1.1 echo database
docker run -it -d -v /database --name db my/centos:v1.1 /bin/bash
```

```
db
[root@bigdata-4 ~]# docker run -d -v /database --name db my/centos:v1.1 echo database
e3e89add4f176c7061aa8c7e3f22f7413103c249be77327d7e6e502326c0a541
[root@bigdata-4 ~]# docker ps
```

在其他容器中使用--volumes-from挂载db容器中的数据卷

```
docker run -it -d --volumes-from db --name db1 -it my/centos:v1.1 /bin/bash
```

```
[root@bigdata-4 ~]# docker run -it -d --volumes-from db --name db1 -it my/centos:v1.1 /bin/bash
6b2e31472b49ce4aa8ab1fae6e7415af712985e37b64d49ba0415e5a4b52b6c
[root@bigdata-4 ~]# docker attach db1
[root@6b2e31472b49 /]# ls
anaconda-post.log  bin  database  dev  etc  home  lib  lib64  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
[root@6b2e31472b49 /]# cd /database/
[root@6b2e31472b49 database]# ls
[root@6b2e31472b49 database]#
```

```
exit
[root@bigdata-4 ~]# docker run -it -d --volumes-from db --name db2 -it my/centos:v1.1 /bin/bash
638bcb42ed86d121f4d72063e48d3301808ed099a3e1857b8e89c3c170366448
[root@bigdata-4 ~]# docker attach db2
[root@638bcb42ed86 /]# ll
total 12
-rw-r--r--   1 root root 12082 Mar  5 17:36 anaconda-post.log
lrwxrwxrwx   1 root root     7 Mar  5 17:34 bin -> usr/bin
drwxr-xr-x   2 root root     6 Aug 13 08:02 database
drwxr-xr-x   5 root root   360 Aug 13 08:16 dev
drwxr-xr-x   1 root root    66 Aug 13 08:16 etc
drwxr-xr-x   2 root root     6 Apr 11  2018 home
lrwxrwxrwx   1 root root     7 Mar  5 17:34 lib -> usr/lib
lrwxrwxrwx   1 root root     9 Mar  5 17:34 lib64 -> usr/lib64
drwxr-xr-x   2 root root     6 Apr 11  2018 media
drwxr-xr-x   2 root root     6 Apr 11  2018 mnt
drwxr-xr-x   2 root root     6 Apr 11  2018 opt
dr-xr-xr-x 125 root root     0 Aug 13 08:16 proc
dr-xr-x---   3 root root   147 Aug 12 04:08 root
drwxr-xr-x  12 root root   163 Aug 12 04:02 run
lrwxrwxrwx   1 root root     8 Mar  5 17:34 sbin -> usr/sbin
drwxr-xr-x   2 root root     6 Apr 11  2018 srv
dr-xr-xr-x  13 root root     0 Aug 13 03:36 sys
drwxrwxrwt   7 root root   132 Aug 12 04:02 tmp
drwxr-xr-x  13 root root   155 Mar  5 17:34 usr
drwxr-xr-x  18 root root   238 Mar  5 17:34 var
[root@638bcb42ed86 /]#
```

可以看到数据卷容器，在多个容器中挂载之后是数据共享的。

```
[root@bigdata-4 ~]# docker attach db
[root@5e452691d1d6 /]# ls
anaconda-post.log  bin  database  dev  etc  home  lib  lib64  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
[root@5e452691d1d6 /]# cd database/
[root@5e452691d1d6 database]# ls
1.txt
[root@5e452691d1d6 database]# echo >> 1.txt
[root@5e452691d1d6 database]# cat 1.txt
fdljfldajfldj
hflkdjfld;ajf

[root@5e452691d1d6 database]# vi 1.txt
[root@5e452691d1d6 database]#
```

```
-rw-r--r--.  1 root     root         31409 May 24 22:04 zookeeper.out
[root@bigdata-4 ~]# docker attach db1
[root@246759b77f62 /]# ls
anaconda-post.log  bin  database  dev  etc  home  lib  lib64  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
[root@246759b77f62 /]# cd database/
[root@246759b77f62 database]# ls
1.txt
[root@246759b77f62 database]# cat 1.txt
fdljfldajfldj
hflkdjfld;ajf
11111111111111111
1111111111111111
[root@246759b77f62 database]#
```

5.数据卷不会被自动删除，要删除一个数据卷，必须在删除最后一个挂着该数据卷的容器，如下

```
docker volume rm name
```

查看数据卷的名称

```
[root@bigdata-4 ~]# docker inspect db
[
    {
        "Id": "0bffa460c9185c0f8cf3b1533b12244d9edb1b2efecd0f4a591b7114bc46a339",
        "Created": "2019-08-13T13:23:02.228162167Z",
        "Path": "/bin/bash",
        "Args": [],
        "State": {
            "Status": "running",
            "Running": true,
            "Paused": false,
"Mounts": [
    {
        "Type": "volume",
        "Name": "31292340bc2b9e6219c1056a656a2f6a74aa85e465885a79a717bb65a0f53c44",
        "Source": "/var/lib/docker/volumes/31292340bc2b9e6219c1056a656a2f6a74aa85e465885a79a717bb65a0f53c44/_data",
        "Destination": "/database",
        "Driver": "local",
        "Mode": "",
        "RW": true,
        "Propagation": ""
    }
]
[root@bigdata-4 ~]# docker volume rm  31292340bc2b9e6219c1056a656a2f6a74aa85e465885a79a717bb65a0f53c44
Error response from daemon: Unable to remove volume, volume still in use: remove 31292340bc2b9e6219c1056a656a2f6a74aa85e465885a79a717bb65a0f53c44: volume is in use - [0bffa460c9185c0f8cf3b1533b12244d9edb1b2efecd0f4a591b7114bc46a339, 2c8e815a5b7aa47f2b1b96a95bd66b58271fa76386858a04f60df2bb92438d37]
[root@bigdata-4 ~]#
```

要想删除数据卷，必须删除所有挂载的容器

```
[root@bigdata-4 ~]# docker rm db1
db1
[root@bigdata-4 ~]# docker rm db
Error response from daemon: You cannot remove a running container 0bffa460c9185c0f8cf3b1533b12244d9edb1b2efecd0f4a591b7114bc46a339. Stop the container before attempting removal or use -f
[root@bigdata-4 ~]# docker rm -f db
db
[root@bigdata-4 ~]#
```

```
[root@bigdata-4 ~]# docker volume rm  31292340bc2b9e6219c1056a656a2f6a74aa85e465885a79a717bb65a0f53c44
Error response from daemon: Unable to remove volume, volume still in use: remove 31292340bc2b9e6219c1056a656a2f6a74aa85e465885a79a717bb65a0f53c44: volume is in use - [0bffa460c9185c0f8cf3b1533b12244d9edb1b2efecd0f4a591b7114bc46a339]
[root@bigdata-4 ~]# docker volume rm  31292340bc2b9e6219c1056a656a2f6a74aa85e465885a79a717bb65a0f53c44
31292340bc2b9e6219c1056a656a2f6a74aa85e465885a79a717bb65a0f53c44
[root@bigdata-4 ~]#
```