|  |
| --- |
| **Docker数据管理** |
|  |
| **版本：V1.0** |

|  |  |  |  |
| --- | --- | --- | --- |
| 拟制人: | 王先健 | 日期: | 2017-6-26 |
| 审核人: |  | 日期: |  |
| 批准人: |  | 日期: |  |

**南京国通智能科技有限公司**

南京国通智能科技有限公司对本文件资料享有著作权及其它专属权利，未经书面许可，不得将该等文件资料（其全部或任何部分）披露予任何第三方，或进行修改后使用。

**修订记录**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **版本号** | **作者** | **修订日期** | **修订内容** | **批准人** |
| 1.0 | 王先健 | 2017-6-26 | 初稿形成 | - |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**目 录**

目录

[一、数据卷 4](#_Toc486255837)

[1. 1、数据卷的添加 4](#_Toc486255838)

[1.2、挂载本地目录到容器目录 5](#_Toc486255839)

[二、数据容器卷 7](#_Toc486255840)

[2.1、创建容器卷并挂载 7](#_Toc486255841)

[2.2、使用--volumes-from挂载容器数据卷 7](#_Toc486255842)

[2.3、容器挂载多个数据卷 9](#_Toc486255843)

[三、利用数据卷容器来迁移数据 9](#_Toc486255844)

[3.1、备份 9](#_Toc486255845)

[3.2、恢复 10](#_Toc486255846)

# 一、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(unix文件系统)，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

● 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。

# 1、数据卷的添加

数据卷的添加可以通过-v 参数来设定，后边跟上目录。

在一次 run 中多次使用可以挂载多个数据卷.一下举例说明：

[root@localhost ~]# mkdir gtintel

[root@localhost ~]# cd gtintel/

[root@localhost gtintel]# ll

total 0

[root@localhost gtintel]# touch test

[root@localhost gtintel]# docker run -d -ti --name gt -v /gtintel docker.io/ubuntu

018b872c68737f64e4e2a014eb83ba32892f2de2bed4d6656a0fea8e776efd25

[root@localhost gtintel]# docker exec -ti gt /bin/bash

root@018b872c6873:/# cd gtintel/

root@018b872c6873:/gtintel# ll

total 0

drwxr-xr-x. 2 root root 6 Apr 17 03:21 ./

drwxr-xr-x. 22 root root 257 Apr 17 03:21 ../

root@018b872c6873:/gtintel# touch abc

# 1.2、挂载本地目录到容器目录

例、挂载本地服务器上的一个目录/mnt到容器test 目录/data中

[root@localhost ~]# docker rm $(docker ps -a -q) #先删除所有后台运行的容器

[root@localhost ~]# docker run -d -ti --name test -v /mnt:/data docker.io/centos

a9a539ad3584b05d76656b082237ad6ec6038f168df79246a2cbaebe941b067f

[root@localhost ~]# docker exec -ti test /bin/bash

[root@a9a539ad3584 /]# cd data/

[root@a9a539ad3584 data]# ll

ls: cannot open directory .: Permission denied #解决方法如下：在宿主机上关闭selinux

[root@a9a539ad3584 data]# ll #此目录下的文件即宿主目录/mnt的内容

total 252232

-rw-r--r--. 1 root root 546 Mar 21 02:25 Dockerfile

-rw-r--r--. 1 root root 12773688 Apr 11 03:19 galera-3-25.3.20-2.el7.x86\_64.rpm

-rw-r--r--. 1 root root 181367942 May 16 2016 jdk-8u91-linux-x64.tar.gz

drwxr-xr-x. 8 10 143 255 Apr 1 2016 jdk1.8.0\_91

-rw-r--r--. 1 root root 127 Mar 20 09:05 mongodb.repo

-rw-r--r--. 1 root root 33320 Apr 11 03:09 mysql-wsrep-5.6-5.6.35-25.18.20170106.1f9ae89.el7.x86\_64.rpm

-rw-r--r--. 1 root root 6236120 Apr 11 03:12 mysql-wsrep-client-5.6-5.6.35-25.18.20170106.1f9ae89.el7.x86\_64.rpm

-rw-r--r--. 1 root root 981708 Apr 11 03:10 mysql-wsrep-devel-5.6-5.6.35-25.18.20170106.1f9ae89.el7.x86\_64.rpm

-rw-r--r--. 1 root root 20169676 Apr 11 03:15 mysql-wsrep-server-5.6-5.6.35-25.18.20170106.1f9ae89.el7.x86\_64.rpm

-rw-r--r--. 1 root root 1287840 Apr 11 03:10 mysql-wsrep-shared-5.6-5.6.35-25.18.20170106.1f9ae89.el7.x86\_64.rpm

drwxr-xr-x. 10 1000 1000 4096 Feb 20 2014 zookeeper-3.4.6

-rw-r--r--. 1 root root 17699306 Jan 10 2016 zookeeper-3.4.6.tar.gz

-rw-r--r--. 1 root root 17699306 Jan 10 2016 zookeeper-3.4.6.tar.gz.1

注意：删除容器的时候，数据卷不会删除。如果要删除容器的时候同时删除数据卷，需加上-v参数。比如: docker rm test -v /data

# 二、数据容器卷

如果用户需要在多个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，数据卷容器也是一个容器，她专门用来提供数据卷供其他容器挂载

## 2.1、创建容器卷并挂载

例、创建数据卷容器dbdata，并在其中建立一个数据卷挂载到/dbdata：

[root@promote ~]# docker run -t -i -v /dbdata --name dbdata ubuntu

root@402198b81ef0:/# ls

bin boot dbdata dev etc home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var

## 2.2、使用--volumes-from挂载容器数据卷

可以在其他的容器中使用--volumes-from来挂载dbdata容器中的数据卷，比如创建test1、test2两个容器，并从dbdata容器挂载数据卷：

[root@promote ~]# docker run -ti --volumes-from dbdata --name test1 ubuntu

[root@promote ~]# docker run -ti --volumes-from dbdata --name test2 ubuntu

在dbdata中新建一个文件hello

root@402198b81ef0:/# cd dbdata/

root@402198b81ef0:/dbdata# touch hello

root@402198b81ef0:/dbdata# ll

total 0

drwxr-xr-x. 2 root root 19 Apr 24 09:11 ./

drwxr-xr-x. 22 root root 256 Apr 24 09:04 ../

-rw-r--r--. 1 root root 0 Apr 24 09:11 hello

root@402198b81ef0:/dbdata#

在test1和test2分别查看是否存在hello文件

test1中显示如下

root@27d3acd7e324:/dbdata# ll

total 0

drwxr-xr-x. 2 root root 19 Apr 24 09:11 ./

drwxr-xr-x. 22 root root 256 Apr 24 09:08 ../

-rw-r--r--. 1 root root 0 Apr 24 09:11 hello

root@27d3acd7e324:/dbdata#

test2中显示如下

[root@promote ~]# docker run -ti --volumes-from dbdata --name test2 ubuntu

root@980a36fdd7ac:/# cd dbdata/

root@980a36fdd7ac:/dbdata# ll

total 0

drwxr-xr-x. 2 root root 19 Apr 24 09:11 ./

drwxr-xr-x. 22 root root 256 Apr 24 09:16 ../

-rw-r--r--. 1 root root 0 Apr 24 09:11 hello

root@980a36fdd7ac:/dbdata#

## 2.3、容器挂载多个数据卷

可以多次使用--volumes-from来从多个容器挂载多个数据卷，也可以从其他已经挂载了容器卷的容器来挂载数据卷

使用--volumes-from参数所挂载的数据卷容器自身并不需要保持在运行状态

[root@promote ~]# docker run -ti --name test3 --volumes-from test1 java

root@14eaf038627d:/# cd dbdata/

root@14eaf038627d:/dbdata# ll

bash: ll: command not found

root@14eaf038627d:/dbdata# ls

hello

root@14eaf038627d:/dbdata#

即使你删除所有挂载了数据卷dbdata的容器（包括最初的dbdata容器和后续的test1和test2），数据卷本身也不会被删除。要删在磁盘上删除这个数据卷，

只能针对最后一个挂载了数据卷的容器显式地调用docker rm -v命令。这种方式可使你在容器之间方便的更新和迁移数据。

# 三、利用数据卷容器来迁移数据

## 3.1、备份

[root@promote ~]# docker run --volumes-from dbdata -v $(pwd):/backup --name worker ubuntu tar cvf /backup/backup.tar /dbdat

首先利用Ubuntu镜像创建了一个worker容器，用--volumes-from dbdata参数使worker容器挂载dbdata容器的数据卷

使用-v ${pwd}:/backup 参数来挂载本地的当前目录到worker容器的/backup目录worker启动后使用tar 命令将/dbdata下的内容备份到/backup

目录下。

## 3.2、恢复

可以在同一容器或在另外的容器中恢复此数据。创建一个新的容器

[root@promote ~]# docker run -v /dbdata --name dbdata2 ubuntu /bin/bash

然后在新的容器中的数据卷里un-tar此备份文件。

[root@promote ~]# docker run --volumes-from dbdata2 -v $(pwd):/backup busybox tar xvf /backup/backup.tar