

ANURAG PACHAURI

R171218026

500069505

Experiment -2

Docker Volume

Volume is simply a directory inside our container. Firstly, We have to declare this as a volume and then share Volume.

Even if we stop container, still we can access volume. Volume will be created in one container. We can declare a directory as a volume only while creating Container. You can't Create Volume from existing container. You can share one volume across any number of container. Volume will not be included when you update an image and make a new container of that image. Volume can be mapped container to container and Host to container and vice-versa.

Benefits:

- Ø Decoupling Containers from storage.
- Ø Share Volume among different Containers.
- Ø Attach Volume to Containers.
- Ø On deleting containers volume does not delete.

```

vagrant@ubuntu-xenial:~
'docker' is not recognized as an internal or external command,
operable program or batch file.

A:\vm2>vagrant up
Bringing machine 'default' up with 'virtualbox' provider...
==> default: Checking if box 'ubuntu/xenial64' version '20201102.0.0' is up to date...
==> default: Machine already provisioned. Run `vagrant provision` or use the `--provision`
==> default: flag to force provisioning. Provisioners marked to run always will still run.

A:\vm2>vagrant ssh
Welcome to Ubuntu 16.04.7 LTS (GNU/Linux 4.4.0-194-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

17 packages can be updated.
0 updates are security updates.

New release '18.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

*** System restart required ***
Last login: Mon Feb  1 09:48:52 2021 from 10.0.2.2
vagrant@ubuntu-xenial:~$ docker --version
Docker version 19.03.13, build 4484c46d9d
vagrant@ubuntu-xenial:~$ docker volume ls
DRIVER          VOLUME NAME
local           myvol
vagrant@ubuntu-xenial:~$ docker run -it alpine
/ # ls
bin    dev    etc    home   lib    media  mnt    opt    proc   root   run    sbin   srv    sys    tmp    usr    var
/ # cd mnt
/mnt # touch abcd.txt
/mnt # touch a.txt
/mnt # ls
a.txt    abcd.txt
/mnt # exit
dvagrant@ubuntu-xenial:~$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS              NAMES
b8f34d5d39f2        alpine              "/bin/sh"           9 minutes ago       Up 9 minutes                   bold_varahamihira
vagrant@ubuntu-xenial:~$ docker run -it alpine
/ # cd mnt
/mnt # ls
/mnt # exit
vagrant@ubuntu-xenial:~$

```

1. Creating and running volume in docker

```
vagrant@ubuntu-xenial: ~
/ # ls
bin    dev    etc    home   lib    media  mnt    opt    proc   root   run    sbin   srv    sys    tmp    usr    var
/ # cd mnt
/mnt # ls
/mnt # touch a.txt
/mnt # touch abcd.txt
/mnt # exit
vagrant@ubuntu-xenial:~$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS              NAMES
b8f34d5d39f2       alpine             "/bin/sh"           13 minutes ago      Up 13 minutes
vagrant@ubuntu-xenial:~$ docker run -it -v myvol2:/mnt alpine
/ # cd mnt
/mnt # ls
a.txt    abcd.txt
/mnt # exit
vagrant@ubuntu-xenial:~$ docker run -it alpine
/ # cd mnt
/mnt # ls
/mnt # exit
vagrant@ubuntu-xenial:~$ docker volume ls
DRIVER              VOLUME NAME
local               myvol
local               myvol2
vagrant@ubuntu-xenial:~$ docker volume inspect myvol2
[
  {
    "CreatedAt": "2021-02-01T10:15:06Z",
    "Driver": "local",
    "Labels": null,
    "Mountpoint": "/var/lib/docker/volumes/myvol2/_data",
    "Name": "myvol2",
    "Options": null,
    "Scope": "local"
  }
]
vagrant@ubuntu-xenial:~$ cd /var/lib/docker/volumes/myvol2/_data
-bash: cd: /var/lib/docker/volumes/myvol2/_data: Permission denied
vagrant@ubuntu-xenial:~$ sudo cd /var/lib/docker/volumes/myvol2/_data
sudo: cd: command not found
vagrant@ubuntu-xenial:~$ sudo ls /var/lib/docker/volumes/myvol2/_data
abcd.txt  a.txt
vagrant@ubuntu-xenial:~$
```

2. Creating volume c1

```
vagrant@ubuntu-xenial: ~  
vagrant@ubuntu-xenial:~$ docker run -it -v myvol2:/mnt -name c1 alpine  
unknown shorthand flag: 'n' in -name  
See 'docker run --help'.  
vagrant@ubuntu-xenial:~$ docker run -it -v myvol2:/mnt ---name c1 alpine  
bad flag syntax: ---name  
See 'docker run --help'.  
vagrant@ubuntu-xenial:~$ docker run -it -v myvol2:/mnt --name c1 alpine  
/# ls  
bin    dev    etc    home   lib    media  mnt    opt    proc   root   run    sbin   srv    sys    tmp    usr    var  
/# cd mnt  
/mnt # ls  
a.txt    abcd.txt  
/mnt # ls  
a.txt    abcd.txt  c.txt  
/mnt #
```

3. Creating volume c2 and seeing the files created in another volume exists there

```
vagrant@ubuntu-xenial: ~  
New release '18.04.5 LTS' available.  
Run 'do-release-upgrade' to upgrade to it.  
  
*** System restart required ***  
Last login: Mon Feb  1 10:21:05 2021 from 10.0.2.2  
0 updates are security updates.  
  
New release '18.04.5 LTS' available.  
Run 'do-release-upgrade' to upgrade to it.  
  
*** System restart required ***  
Last login: Mon Feb  1 10:21:05 2021 from 10.0.2.2  
vagrant@ubuntu-xenial:~$ docker ps  
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS              NAMES  
76db5868b5ab       alpine             "/bin/sh"          6 minutes ago       Up 6 minutes                c1  
b8f34d5d39f2       alpine             "/bin/sh"          24 minutes ago      Up 24 minutes                bold_varahamihira  
vagrant@ubuntu-xenial:~$ docker run -it -v myvol2:/mnt --name c2 alpine  
docker: Error response from daemon: Conflict. The container name "/c2" is already in use by container "3a8f25ea77001e2758e132a6c03974c205d". You can reuse the name with the -f flag.  
See 'docker run --help'.  
vagrant@ubuntu-xenial:~$ docker run -it -v myvol2:/mnt --name c3 alpine  
/ # cd /mnt  
/mnt # ls  
a.txt      abcd.txt  
/mnt # touch c.txt  
/mnt # ls  
a.txt      abcd.txt  c.txt  
/mnt #
```



Type here to search

