

Experiment 8

Aim

To share data between containers using Dockervolumes

Theory

Introduced in Docker's 1.9 release, the `docker volume create` command allows you to create a volume without relating it to any particular container. We'll use this command to add a volume named `DataVolume1`

Learnt it the hard way that if you are on old linux kernel, the runc has a weird issue that leads to exploitation in docker which can grant you root access on host machine which was fixed and the fix legit broke docker on various places

- To create the volume:

```
docker volume create --name DataVolume1
```



The screenshot shows a terminal window with the following commands and output:

```
~/.Partition/BooksPirated
> goto projects

~/.SoopaProject
> ls
'11-ecology notes PDF.pdf'  'Compiler design'  gfg  go  'Maths Research'  Resume
AI  configurations  go  got  MPICP
ai-chat-bot  crawling  got  got  nohup.out  Reversing
AI-chat-bot  debug  'Hacker's Delight 2nd Edition.pdf'  Onworks  Router
AI.zip  development  Hacking  'Operating Systems Portfolio.html'  roxor-arcade
alphacoders  devops  Hadoop  os  SpyderBot
ansible  docker  'Important books'  PacketTracer  Stalking
ArtificialIntelligence  'Engineering Hallticket ...pdf'  pev  Philosophy  Streamlocks
Assembly  Environment_Management.pdf  katfetch  php  Tele
Babun  Examples  Kindie  projectEuler  Telegram
Backup  Experiment  KodiLinuxRemote  puppet_v0.4.0_linux_amd64.zip  TelegramProjects
Blog  Experiments  kodilinuxremote  python  Tiny7.iso
BlogIt  file  kokotime  Python  TutorialsPoint
Bot  file2  LectureNotes  PythonApps  'UNIX and Linux System Administration Handbook (5th Ed)(gnv64).pdf'
BruteForce  fnd-win32-v0.0_2b--rev325.7z  log.zip  R  WebServer
C  FUSEfs  MachineLearning  repo

~/.SoopaProject
> cd devops

~/.SoopaProject/devops
> docker volume create --name DataVolume1
DataVolume1

~/.SoopaProject/devops
> docker run -ti --rm -v DataVolume1:/datavolume1 ubuntu
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
5bed26d33975: Pull complete
f11b29a9c730: Pull complete
930bda195c84: Pull complete
78bf95a5d49e: Extracting [=====>] 162B/162B
[/]
```

After that, execute

```
docker run -ti --rm -v DataVolume1:/datavolume1 ubuntu
```

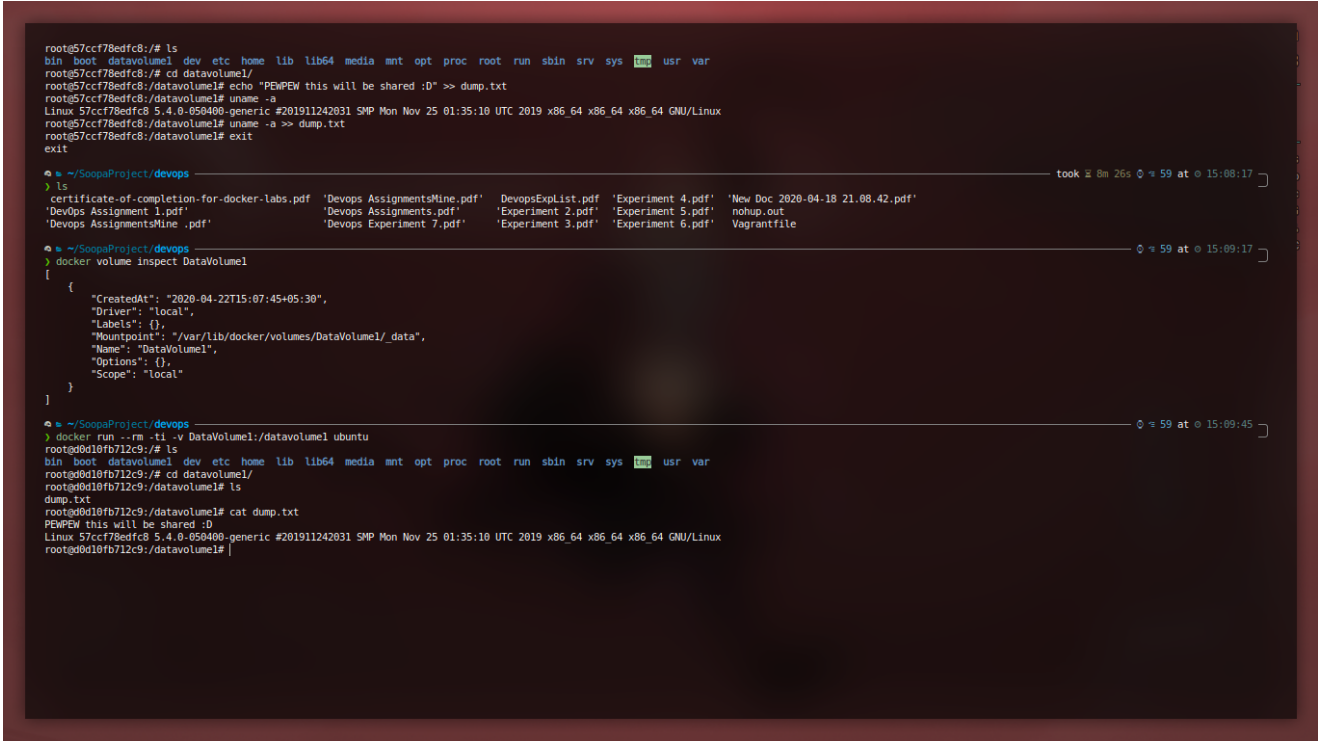
and this will drop you in root shell on ubuntu container

Then go ahead and install some cool stuffs like `figlet` and `sl` , or for simple demonstration lets stick to `echo` :P

```
apt update
apt install figlet sl tree
```

Looking into the directories we find a folder named `datavolume1`, this is basically a common folder between our container and host.

we echo out some stuffs and then simply exit the root shell. After that we go ahead and check the `dataVolume1` and inspect it, now time to mount the second container and view the dumped data into the container.

A terminal window with a dark background and light-colored text. The first part shows a container shell where the user lists files, echoes a message to a file, and exits. The second part shows the host shell where the user lists files, inspects the 'DataVolume1' Docker volume, and then runs a new container that mounts the volume and echoes the same message. The output of the second container shows the message was successfully shared.

```
root@57ccf78edfc8:/# ls
bin boot datavolume1 dev etc home lib lib64 media mnt opt proc root run sbin srv sys usr var
root@57ccf78edfc8:/# cd datavolume1/
root@57ccf78edfc8:/datavolume1# echo "PEWPEW this will be shared :D" >> dump.txt
root@57ccf78edfc8:/datavolume1# uname -a
Linux 57ccf78edfc8 5.4.0-050400-generic #201911242031 SMP Mon Nov 25 01:35:10 UTC 2019 x86_64 x86_64 x86_64 GNU/Linux
root@57ccf78edfc8:/datavolume1# uname -a >> dump.txt
root@57ccf78edfc8:/datavolume1# exit
exit

~ /SoopaProject/devops
> ls
certificate-of-completion-for-docker-labs.pdf 'DevOps AssignmentMine.pdf' DevopsExpList.pdf 'Experiment 4.pdf' 'New Doc 2020-04-18 21.08.42.pdf'
'DevOps Assignment 1.pdf' 'DevOps Assignments.pdf' 'Experiment 2.pdf' 'Experiment 5.pdf' nohup.out
'DevOps AssignmentsMine .pdf' 'DevOps Experiment 7.pdf' 'Experiment 3.pdf' 'Experiment 6.pdf' Vagrantfile

~ /SoopaProject/devops
> docker volume inspect DataVolume1
[
  {
    "CreatedAt": "2020-04-22T15:07:45+05:30",
    "Driver": "local",
    "Labels": {},
    "Mountpoint": "/var/lib/docker/volumes/DataVolume1/_data",
    "Name": "DataVolume1",
    "Options": {},
    "Scope": "local"
  }
]

~ /SoopaProject/devops
> docker run --rm -ti -v DataVolume1:/datavolume1 ubuntu
root@9d18fb712c9:/# ls
bin boot datavolume1 dev etc home lib lib64 media mnt opt proc root run sbin srv sys usr var
root@9d18fb712c9:/# cd datavolume1/
root@9d18fb712c9:/datavolume1# ls
dump.txt
root@9d18fb712c9:/datavolume1# cat dump.txt
PEWPEW this will be shared :D
Linux 57ccf78edfc8 5.4.0-050400-generic #201911242031 SMP Mon Nov 25 01:35:10 UTC 2019 x86_64 x86_64 x86_64 GNU/Linux
root@9d18fb712c9:/datavolume1#
```

As you can see the line "PEWPEW this will be shared has been shared between the containers"

Step 2 — Creating a persistant volume

We'll use the `docker run` command to create a new container using the base Ubuntu image. `-t` will give us a terminal, and `-i` will allow us to interact with it. For clarity, we'll use `--name` to identify the container.

The `-v` flag will allow us to create a new volume, which we'll call `DataVolume2` . We'll use a colon to separate this name from the path where the volume should be mounted in the container

```
docker run -ti --name=Container2 -v DataVolume2:/datavolume2 ubuntu
```

which drops us in root shell and we

```
echo "pewpew this will be shared too :D" >> datavolume2/dump.txt
```

```
root@57ccf78edfc8:/# ls
bin boot datavolume1 dev etc home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var
root@57ccf78edfc8:/# cd datavolume1/
root@57ccf78edfc8:/datavolume1# echo "PEWPEW this will be shared :D" >> dump.txt
root@57ccf78edfc8:/datavolume1# uname -a
Linux 57ccf78edfc8 5.4.0-050400-generic #201911242031 SMP Mon Nov 25 01:35:10 UTC 2019 x86_64 x86_64 x86_64 GNU/Linux
root@57ccf78edfc8:/datavolume1# uname -a >> dump.txt
root@57ccf78edfc8:/datavolume1# exit
exit

❯ ~/SoopaProject/devops took 8m 26s @ 59 at 15:08:17
> ls
certificate-of-completion-for-docker-labs.pdf 'DevOps AssignmentsMine.pdf' DevOpsExpList.pdf 'Experiment 4.pdf' 'New Doc 2020-04-18 21:08:42.pdf'
'DevOps Assignment 1.pdf' 'DevOps Assignments.pdf' 'Experiment 2.pdf' 'Experiment 5.pdf' nohup.out Vagrantfile
'DevOps AssignmentsMine .pdf' 'DevOps Experiment 7.pdf' 'Experiment 3.pdf' 'Experiment 6.pdf'

❯ ~/SoopaProject/devops
> docker volume inspect DataVolume1
[
  {
    "CreatedAt": "2020-04-22T15:07:45+05:30",
    "Driver": "local",
    "Labels": {},
    "Mountpoint": "/var/lib/docker/volumes/DataVolume1/_data",
    "Name": "DataVolume1",
    "Options": {},
    "Scope": "local"
  }
]

❯ ~/SoopaProject/devops
> docker run --rm -ti -v DataVolume1:/datavolume1 ubuntu
root@0d18fb712c9:/# ls
bin boot datavolume1 dev etc home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var
root@0d18fb712c9:/# cd datavolume1/
root@0d18fb712c9:/datavolume1# ls
dump.txt
root@0d18fb712c9:/datavolume1# cat dump.txt
PEWPEW this will be shared :D
Linux 57ccf78edfc8 5.4.0-050400-generic #201911242031 SMP Mon Nov 25 01:35:10 UTC 2019 x86_64 x86_64 x86_64 GNU/Linux
root@0d18fb712c9:/datavolume1#
```

After that we go ahead and,

```
docker start -ai Container2
```

```
❯ ~/SoopaProject/devops took 2m 31s @ 59 at 15:13:08
> docker run -ti --name=Container2 -v DataVolume2:/datavolume2 ubuntu

root@f1790ee4da07:/# ls
bin boot datavolume2 dev etc home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var
root@f1790ee4da07:/# echo "pewpew this will be shared too :D" >> datavolume2/dump.txt
root@f1790ee4da07:/# cat datavolume2/dump.txt
pewpew this will be shared too :D
root@f1790ee4da07:/# exit
exit

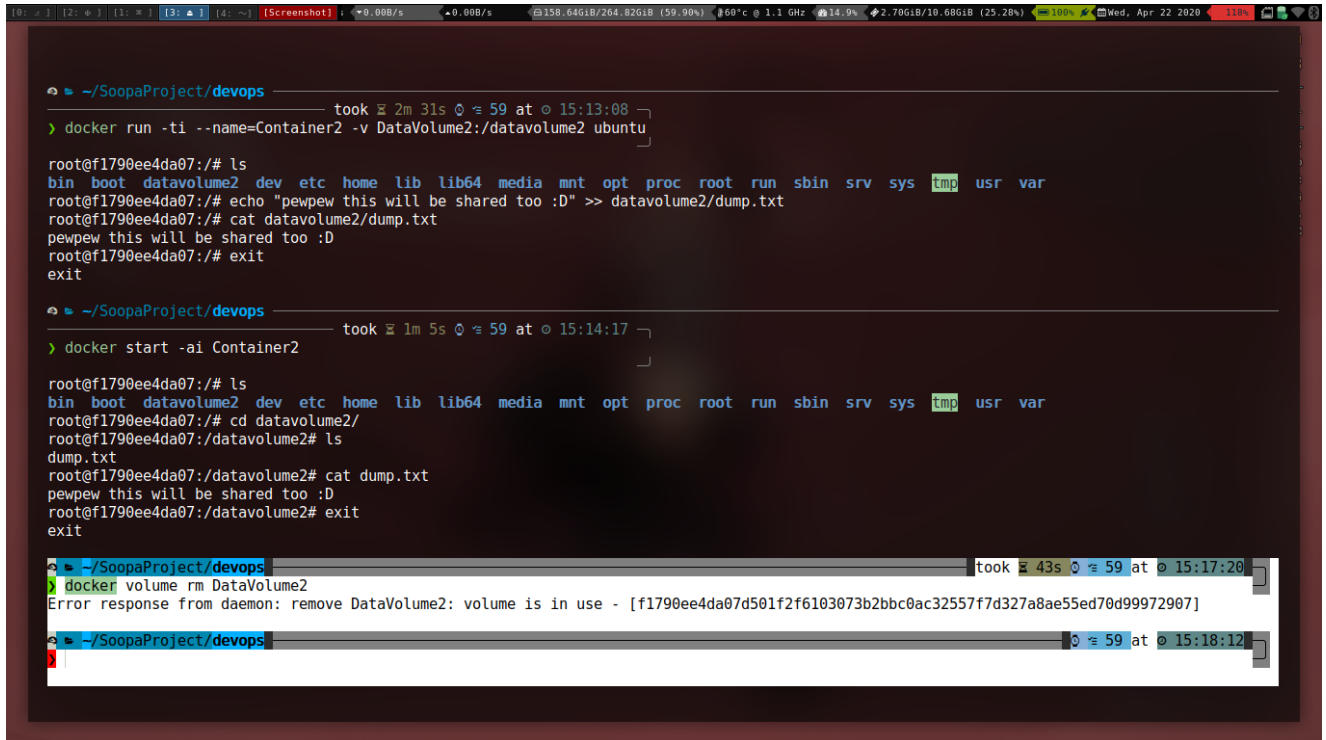
❯ ~/SoopaProject/devops took 1m 5s @ 59 at 15:14:17
> docker start -ai Container2

root@f1790ee4da07:/# ls
bin boot datavolume2 dev etc home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var
root@f1790ee4da07:/# cd datavolume2/
root@f1790ee4da07:/datavolume2# ls
dump.txt
root@f1790ee4da07:/datavolume2# cat dump.txt
pewpew this will be shared too :D
root@f1790ee4da07:/datavolume2#
```

so as to prove persistence, we shall exit the container shell and try to remove the volume while the container is running.

```
docker volume rm DataVolume2
```

We encounter an error.



```
~/SoopaProject/devops took 2m 31s @ 15:13:08
> docker run -ti --name=Container2 -v DataVolume2:/datavolume2 ubuntu

root@f1790ee4da07:/# ls
bin boot datavolume2 dev etc home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var
root@f1790ee4da07:/# echo "pewpew this will be shared too :D" >> datavolume2/dump.txt
root@f1790ee4da07:/# cat datavolume2/dump.txt
pewpew this will be shared too :D
root@f1790ee4da07:/# exit
exit

~/SoopaProject/devops took 1m 5s @ 15:14:17
> docker start -ai Container2

root@f1790ee4da07:/# ls
bin boot datavolume2 dev etc home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var
root@f1790ee4da07:/# cd datavolume2/
root@f1790ee4da07:/datavolume2# ls
dump.txt
root@f1790ee4da07:/datavolume2# cat dump.txt
pewpew this will be shared too :D
root@f1790ee4da07:/datavolume2# exit
exit

~/SoopaProject/devops took 43s @ 15:17:20
> docker volume rm DataVolume2
Error response from daemon: remove DataVolume2: volume is in use - [f1790ee4da07d501f2f6103073b2bbc0ac32557f7d327a8ae55ed70d99972907]
```

We can also create a datavolume from an already existing folder, this is very helpful in cases of log viewing and effective management.

```
docker run -ti --rm -v DataVolume3:/var ubuntu
```

and then exit from the container and run this in ubuntu shell

```
docker run --rm -v DataVolume3:/datavolume3 ubuntu ls datavolume3
```

which gives us,

```
[0: ~] [2: ~] [3: ~] [4: ~] [Screenshot] | 133.00B/s 399.00B/s 158.64GiB/264.82GiB (59.91%) 59°C @ 1.3 GHz 15.3% 2.78GiB/10.68GiB (26.06%) 100% Wed, Apr 22 2020 118%

~/SoopaProject/devops 59 at 15:18:12
> docker run -ti --rm -v DataVolume3:/var ubuntu
root@18d07e7d9f4a:/# ls
bin boot dev etc home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var
root@18d07e7d9f4a:/# cd vat
bash: cd: vat: No such file or directory
root@18d07e7d9f4a:/# cd var
root@18d07e7d9f4a:/var# ls
backups cache lib local lock log mail opt run spool tmp
root@18d07e7d9f4a:/var# exit
exit

~/SoopaProject/devops took 18s 59 at 15:19:59
> docker run --rm -v DataVolume3:/datavolume3 ubuntu ls datavolume3
backups
cache
lib
local
lock
log
mail
opt
run
spool
tmp

~/SoopaProject/devops took 5s 59 at 15:20:36
> |
```

The process of mounting volumes over multiple container is very much same and their removal involves usage of `docker rm` command.

Conclusion

Henceforth, We have successfully finished the study of docker volumes and experimented successfully with them.