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
```

After that, execute

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

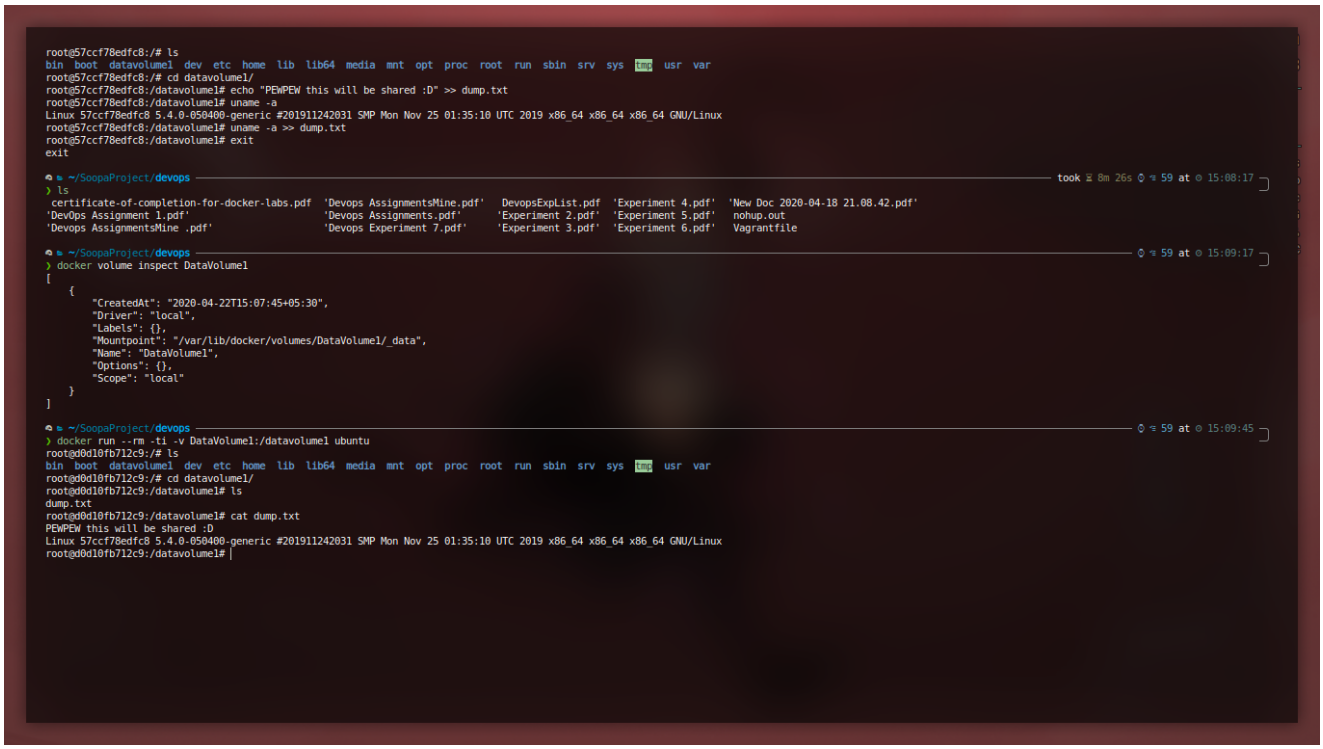
and this will drop you in root shell on ubuntu container

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.



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

Step 2 — Creating a persistant volume

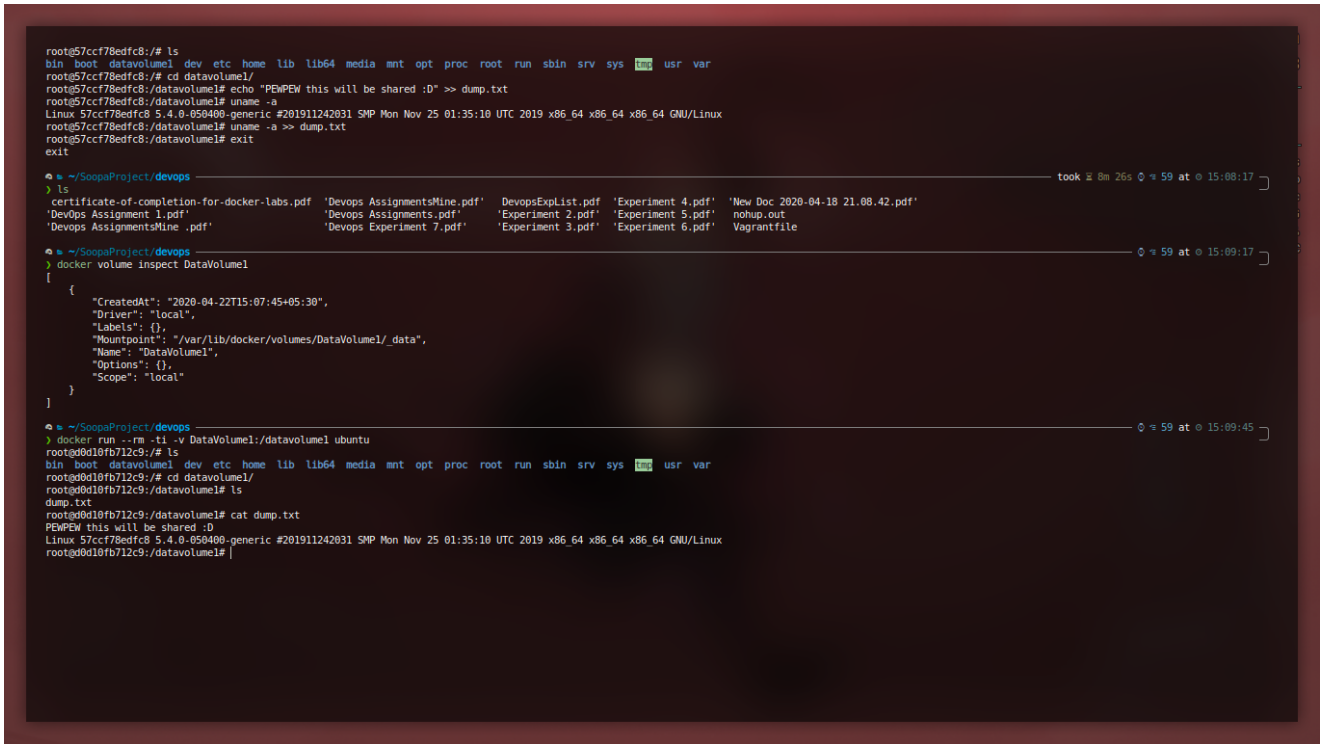
We'll use the `docker run` command to create a new container using the base Ubuntu image. `-t` will give us a terminal, and `-i` will allow us to interact with it. For clarity, we'll use `--name` to identify the container.

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
```

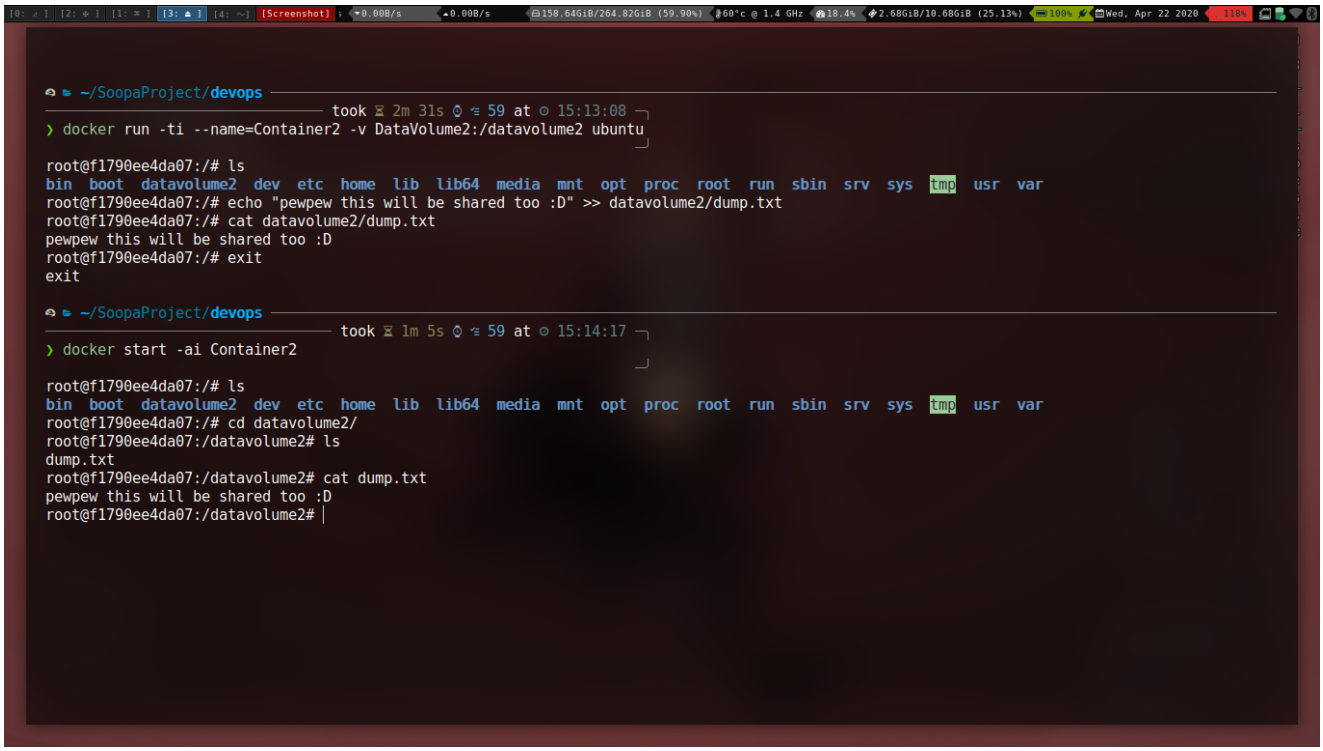
which drops us in root shell and we

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



After that we go ahead and,

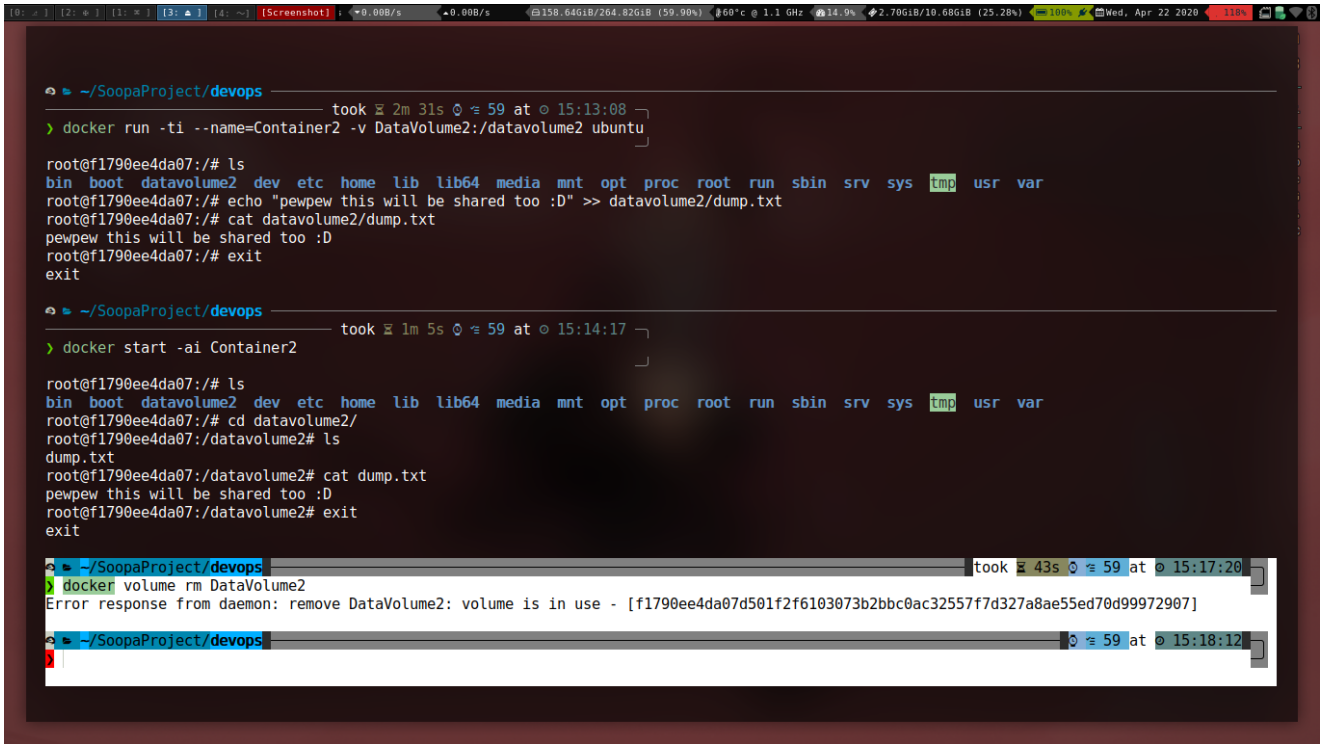
```
docker start -ai Container2
```



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.



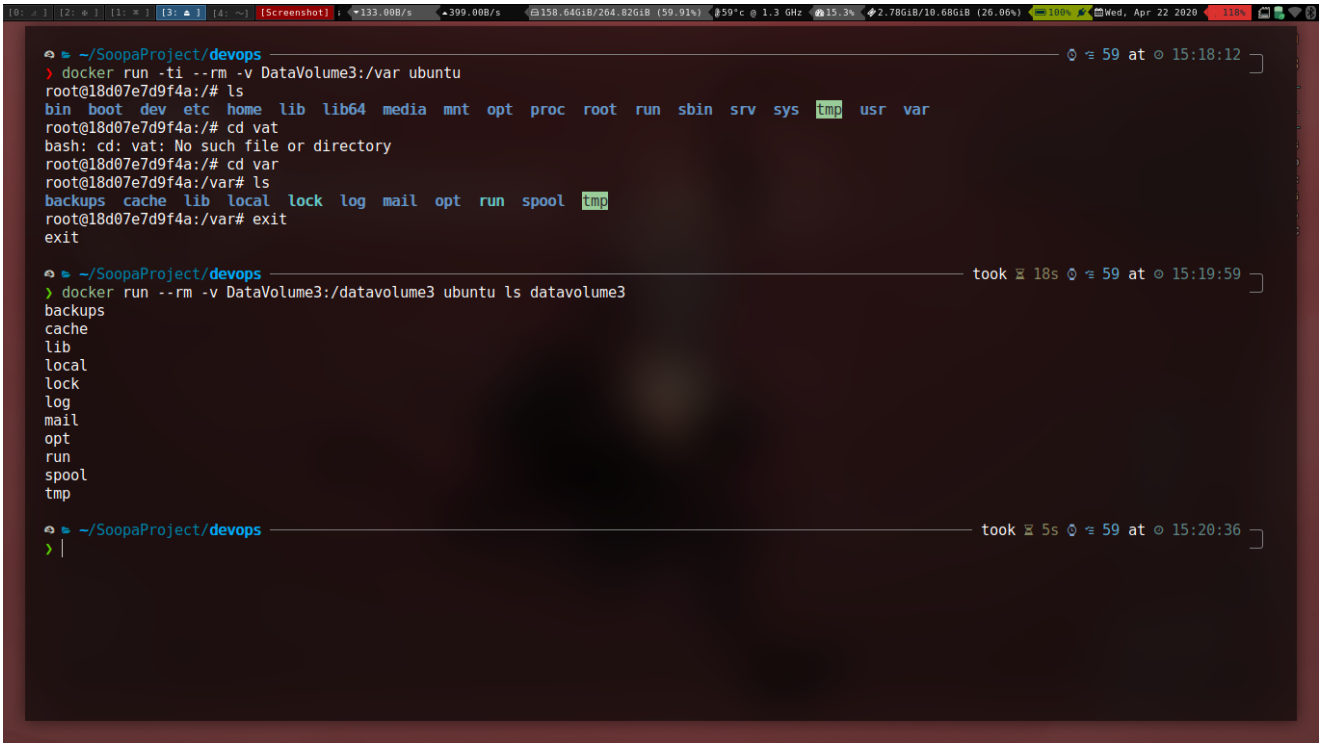
We can also create a datavolume from an already existing folder, this is very helpful in cases of log viewing and effective management.

```
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,



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.