NIPUN SINGAL
500069052
R171218069

Application Containerization Lab

EXPERIMENT-3

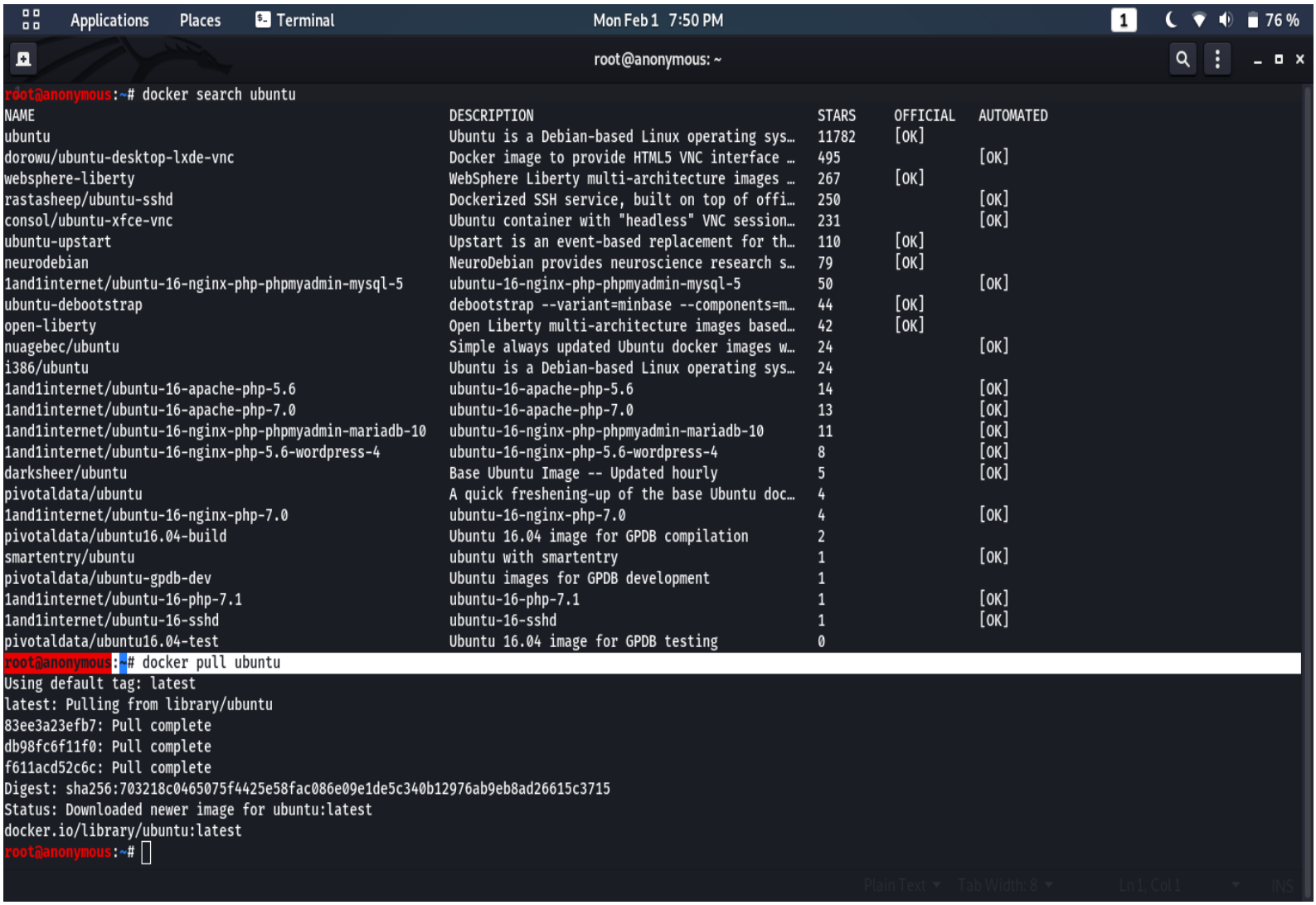
Working with Volumes in docker

Few commands to learn before:

docker search [image-name]
docker pull [image-name]
docker run -it [image-name]
touch [file-name]
docker run -it -v [volume-name]:/[folder-name] [image-name]
docker ps
docker ps -a
docker inspect [volume-name]

Step1: Search and pull a docker image, if already have don't do.

eg: docker search ubuntu
docker pull ubuntu



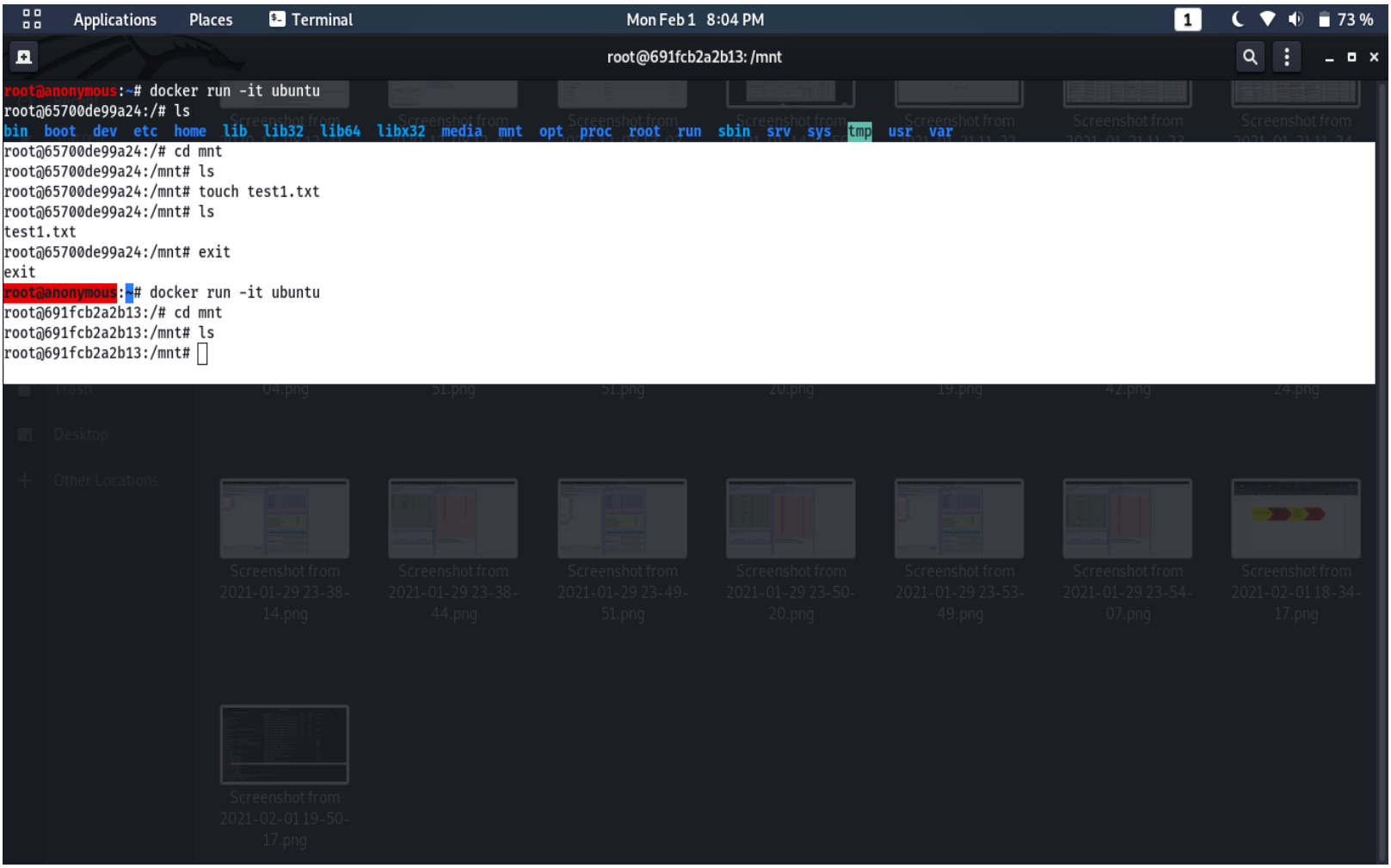
Step2: run the docker image

eg: docker run -it ubuntu

Step3: change the directory to mnt and create a file in that directory and then exit the container.

Eg: **cd mnt**
touch test1.txt
exit

Step4: Now again run the docker image and navigate to mnt directory and do “ls” to check the files. The file you created will not be their.



Step5: Now we will use the concept of volume. Run the docker image with volume option

eg: docker run -it -v volume1:/mnt ubuntu

Step6: Now change the directory to the chosen directory i will take the mnt. Now create a file in it. Do ls you will see the files. Now exit

eg: **cd mnt**
touch test2.txt
touch test2.txt
ls
exit

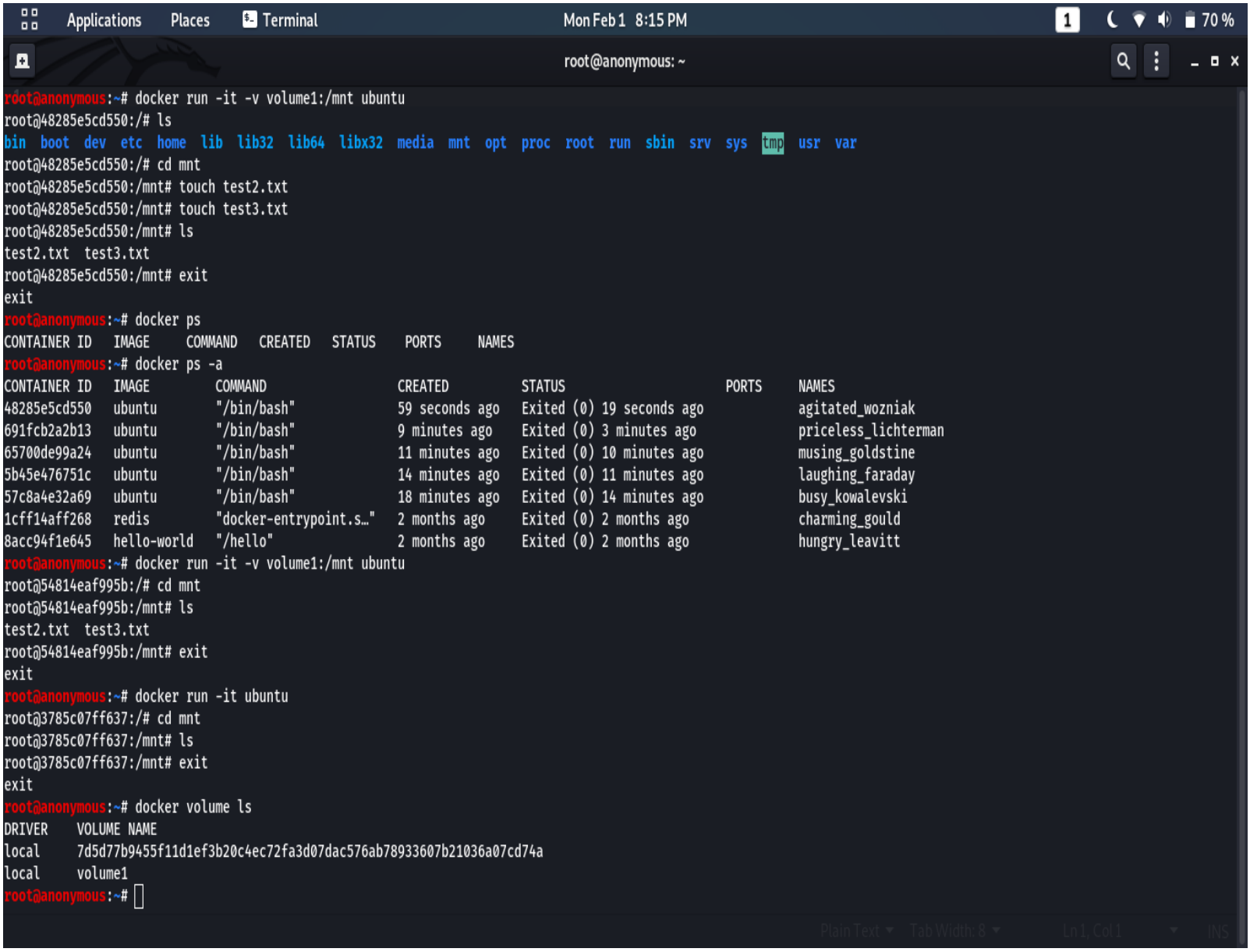
Step7: now run the ubuntu again without volume option and cd to mnt and do ls. You will not see the files you created. Exit it.

Step8: Now again run the ubuntu image with volume option and chose the same volume you created above. Then navigate to mnt folder and do ls. You will see the files that we created before.

eg: docker run -it -v volume1:/mnt ubuntu
cd mnt
ls

This happens because all our files are saved in the volume which is stored in the host machine. So exiting the container will not delete the files

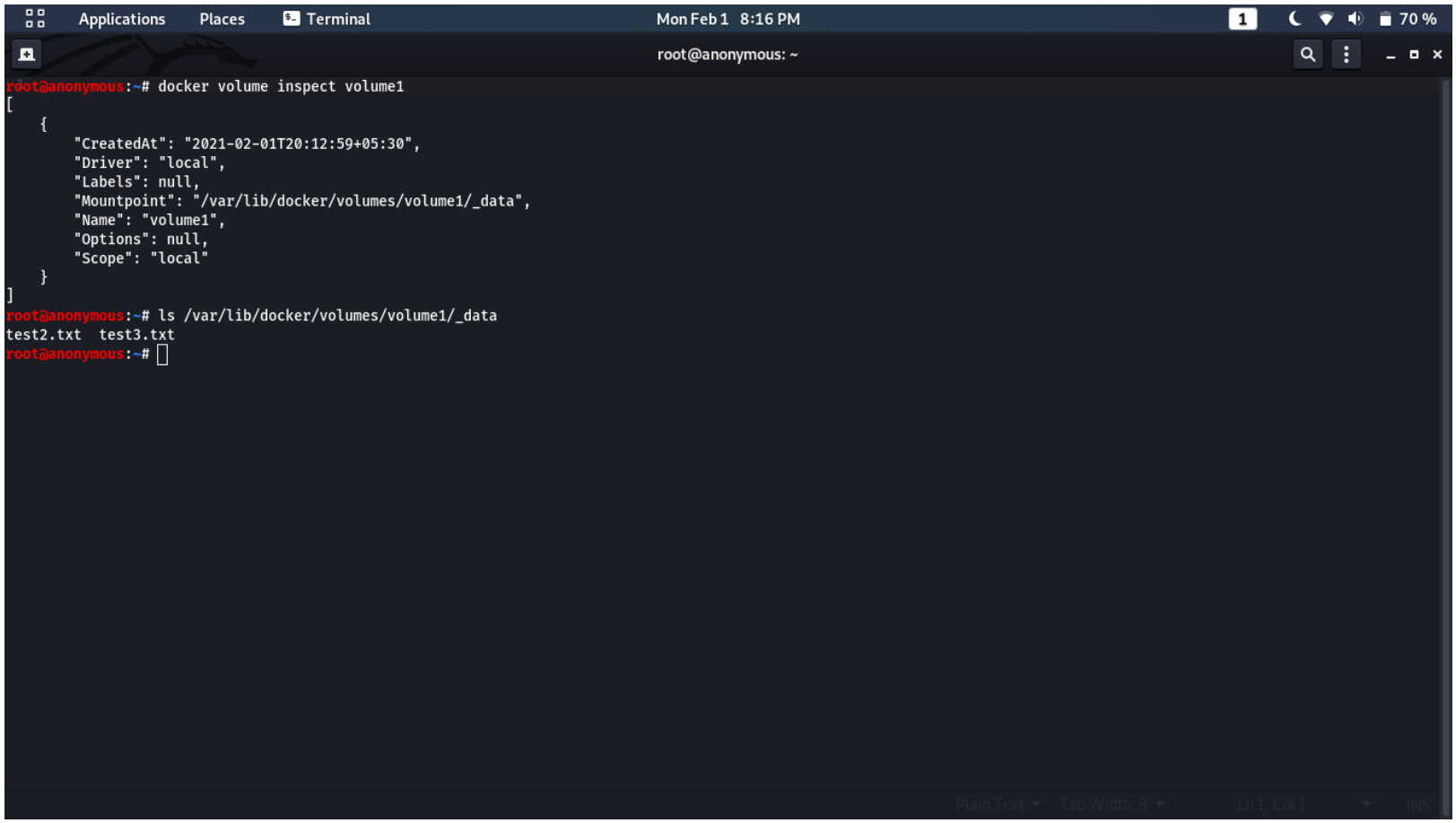
Step9: Do “docker volume ls” to list all the volumes we created.



Step10: We can see the path of the volume/ where our data is getting stored. For this we need to type the command:

eg: docker inspect volume1

Step11: if we do ls to that path we will see our files there.



Step12: Now run the ubuntu image again with the volume option in one terminal and run the alpine image with volume option in the another terminal use different folder in it. Use the same volume.

Eg: docker run -it -v volume1:/mnt ubuntu
cd mnt

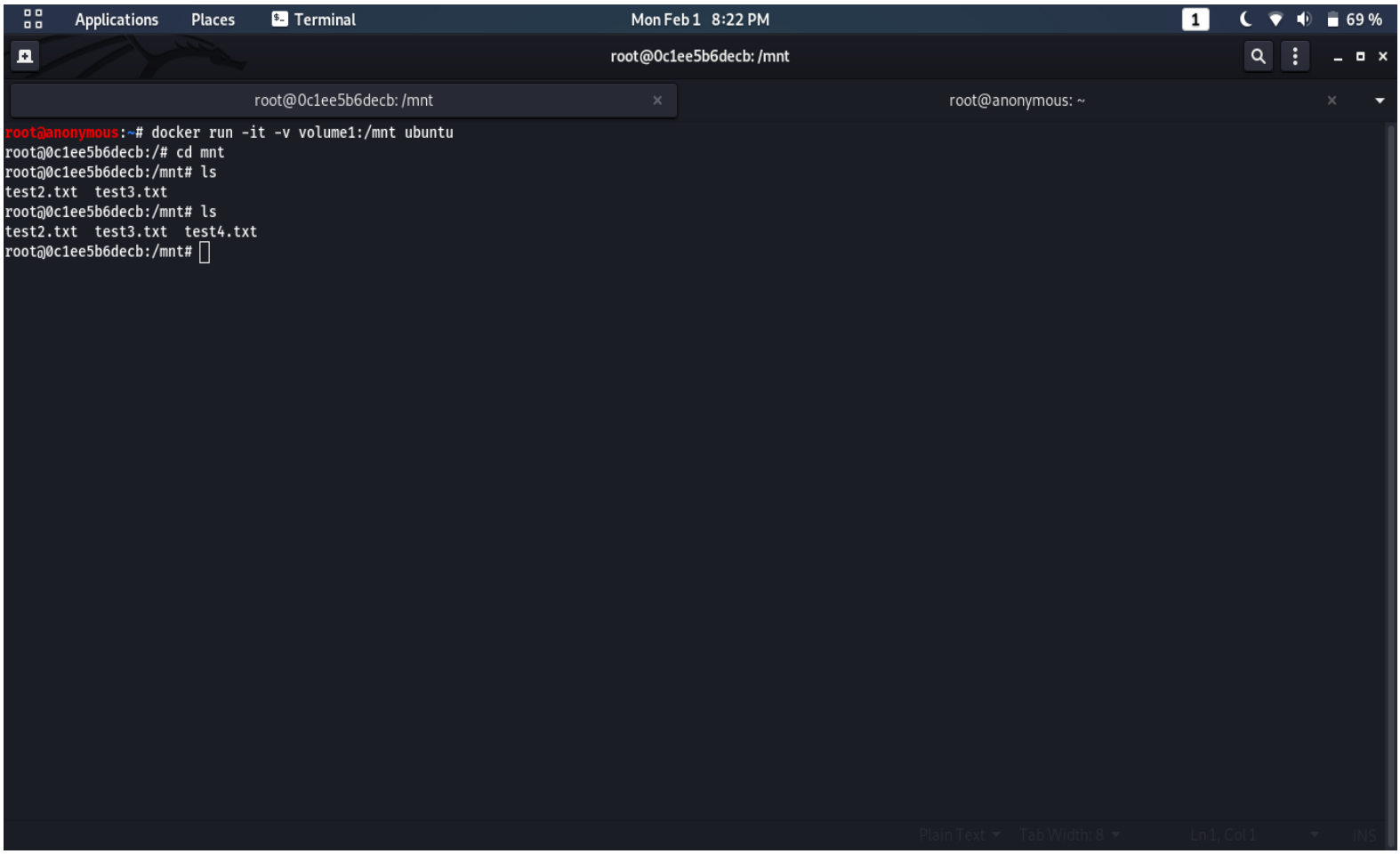
Eg: docker run -it -v volume1:/root alpine
cd root

Step13: now create a file in alpine root folder and do ls in mnt folder of ubuntu. You will see the file that we created in alpine is also visible in mnt folder of ubuntu. **This shows we can we use volume as the file sharing between different containers**

eg: cd root
touch test4.txt
ls

eg: cd mnt
ls


Applications
Places
Terminal
Mon Feb 1 8:23 PM
69 %
root@anonymous: ~
root@0c1ee5b6decb: /mnt
root@anonymous: ~
root@anonymous:~# docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
alpine latest e50c909a8df2 3 days ago 5.61MB
ubuntu latest f63181f19b2f 11 days ago 72.9MB
redis latest 74d107221092 2 months ago 104MB
hello-world latest bf756fb1ae65 13 months ago 13.3kB
root@anonymous:~# docker run -it alpine
/ # ls
bin dev etc home lib media mnt opt proc root run sbin srv sys tmp usr var
/ # cd root
~ # ls
~ # exit
root@anonymous:~# docker run -it -v volume1:/root alpine
/ # cd root
~ # ls
test2.txt test3.txt
~ # touch test4.txt
~ # ld
/bin/sh: ld: not found
~ # ls
test2.txt test3.txt test4.txt
~ #


Applications
Places
Terminal
Mon Feb 1 8:22 PM
69 %
root@0c1ee5b6decb: /mnt
root@0c1ee5b6decb: /mnt
root@anonymous: ~
root@anonymous:~# docker run -it -v volume1:/mnt ubuntu
root@0c1ee5b6decb:/# cd mnt
root@0c1ee5b6decb:/mnt# ls
test2.txt test3.txt
root@0c1ee5b6decb:/mnt# ls
test2.txt test3.txt test4.txt
root@0c1ee5b6decb:/mnt#

Step14: we can delete the volume by using the command:

docker volume rm [volume-name]