

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

```
root@anonymous:~# docker search ubuntu
```

NAME	DESCRIPTION	STARS	OFFICIAL	AUTOMATED
ubuntu	Ubuntu is a Debian-based Linux operating sys...	11782	[OK]	
dorowu/ubuntu-desktop-lxde-vnc	Docker image to provide HTML5 VNC interface ...	495		[OK]
webspHERE-liberty	WebSphere Liberty multi-architecture images ...	267	[OK]	
rastasheep/ubuntu-sshd	Dockerized SSH service, built on top of offi...	250		[OK]
consol/ubuntu-xfce-vnc	Ubuntu container with "headless" VNC session...	231		[OK]
ubuntu-upstart	Upstart is an event-based replacement for th...	110	[OK]	
neurodebian	NeuroDebian provides neuroscience research s...	79	[OK]	
land1internet/ubuntu-16-nginx-php-phpmyadmin-mysql-5	ubuntu-16-nginx-php-phpmyadmin-mysql-5	50		[OK]
ubuntu-debootstrap	debootstrap --variant=minbase --components=m...	44	[OK]	
open-liberty	Open Liberty multi-architecture images based...	42	[OK]	
nuagebec/ubuntu	Simple always updated Ubuntu docker images w...	24		[OK]
i386/ubuntu	Ubuntu is a Debian-based Linux operating sys...	24		
land1internet/ubuntu-16-apache-php-5.6	ubuntu-16-apache-php-5.6	14		[OK]
land1internet/ubuntu-16-apache-php-7.0	ubuntu-16-apache-php-7.0	13		[OK]
land1internet/ubuntu-16-nginx-php-phpmyadmin-mariadb-10	ubuntu-16-nginx-php-phpmyadmin-mariadb-10	11		[OK]
land1internet/ubuntu-16-nginx-php-5.6-wordpress-4	ubuntu-16-nginx-php-5.6-wordpress-4	8		[OK]
darksheer/ubuntu	Base Ubuntu Image -- Updated hourly	5		[OK]
pivotaldata/ubuntu	A quick freshening-up of the base Ubuntu doc...	4		
land1internet/ubuntu-16-nginx-php-7.0	ubuntu-16-nginx-php-7.0	4		[OK]
pivotaldata/ubuntu16.04-build	Ubuntu 16.04 image for GPDB compilation	2		
smartentry/ubuntu	ubuntu with smartentry	1		[OK]
pivotaldata/ubuntu-gpdb-dev	Ubuntu images for GPDB development	1		
land1internet/ubuntu-16-php-7.1	ubuntu-16-php-7.1	1		[OK]
land1internet/ubuntu-16-sshd	ubuntu-16-sshd	1		[OK]
pivotaldata/ubuntu16.04-test	Ubuntu 16.04 image for GPDB testing	0		

```
root@anonymous:~# docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
83ee3a23efb7: Pull complete
db98fc6f11f0: Pull complete
f611acd52c6c: Pull complete
Digest: sha256:703218c0465075f4425e58fac086e09e1de5c340b12976ab9eb8ad26615c3715
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest
root@anonymous:~#
```

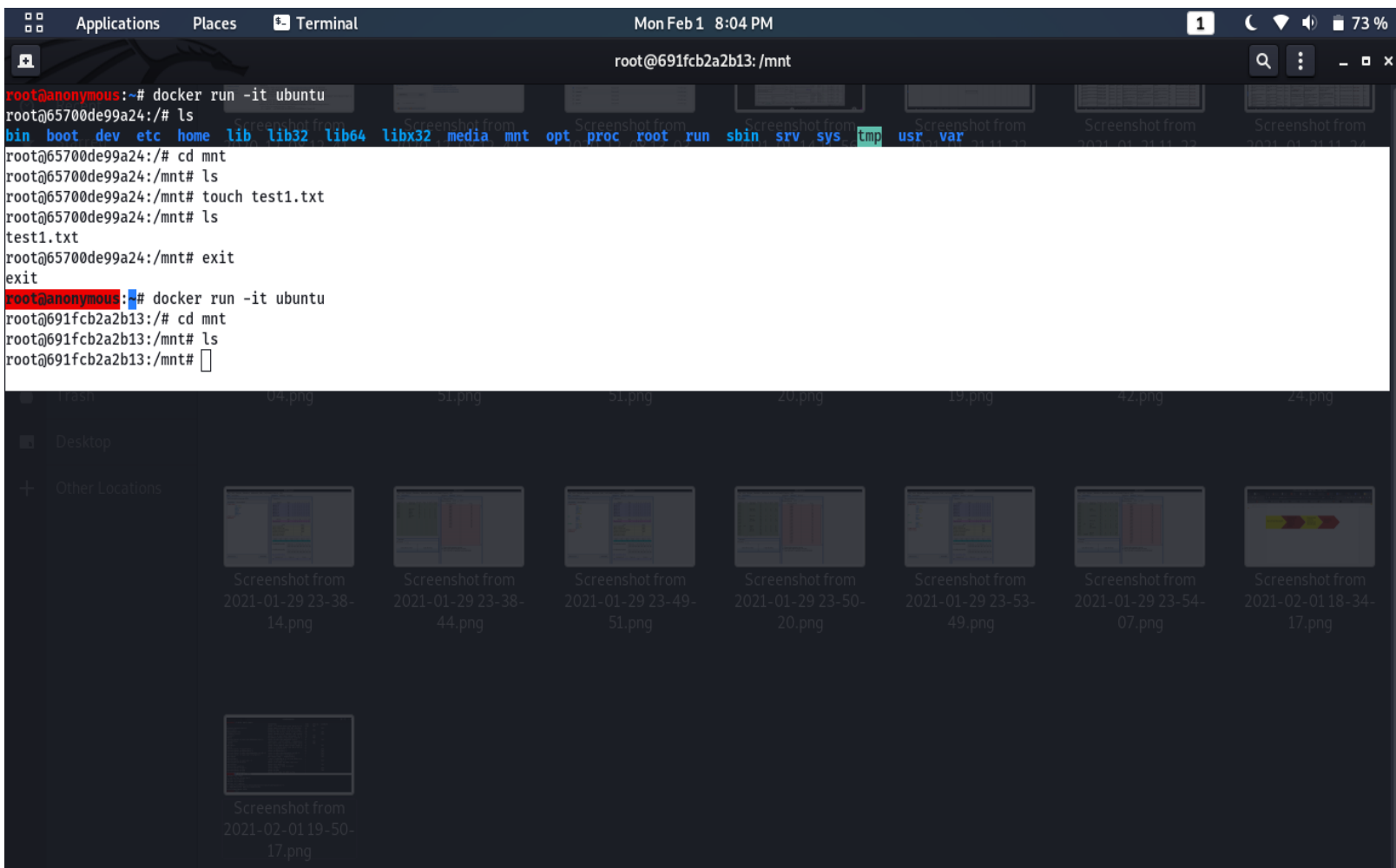
Step2: run the docker image

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.

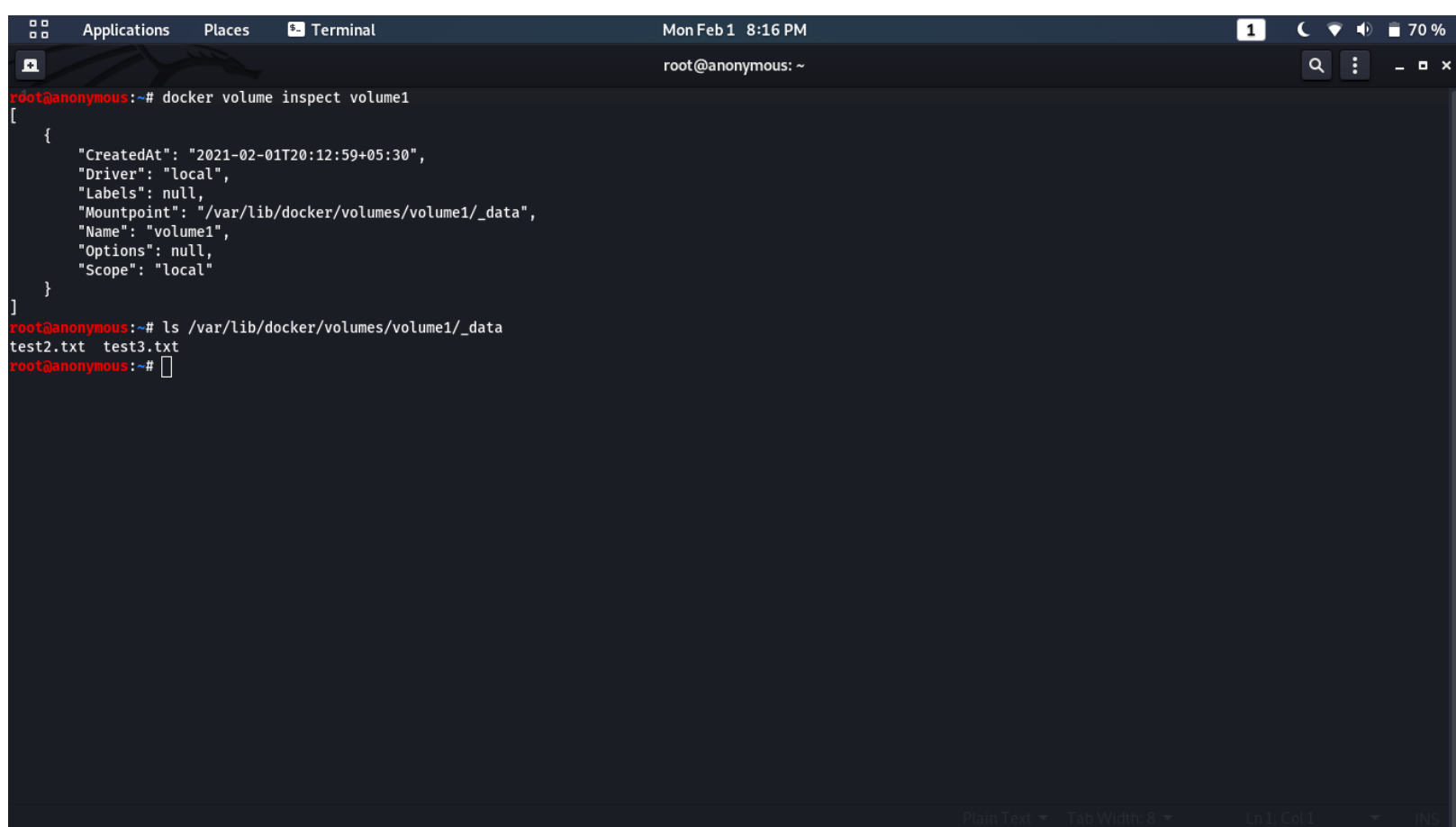
```
Applications  Places  Terminal  Mon Feb 1 8:15 PM  1  70 %
root@anonymous: ~

root@anonymous:~# docker run -it -v volume1:/mnt ubuntu
root@48285e5cd550:/# ls
bin boot dev etc home lib lib32 lib64 libx32 media mnt opt proc root run sbin srv sys tmp usr var
root@48285e5cd550:/# cd mnt
root@48285e5cd550:/mnt# touch test2.txt
root@48285e5cd550:/mnt# touch test3.txt
root@48285e5cd550:/mnt# ls
test2.txt test3.txt
root@48285e5cd550:/mnt# exit
exit
root@anonymous:~# docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
root@anonymous:~# docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED        STATUS       PORTS   NAMES
48285e5cd550   ubuntu   "/bin/bash"   59 seconds ago   Exited (0) 19 seconds ago           agitated_wozniak
691fcb2a2b13   ubuntu   "/bin/bash"   9 minutes ago    Exited (0) 3 minutes ago           priceless_lichterman
65700de99a24   ubuntu   "/bin/bash"   11 minutes ago   Exited (0) 10 minutes ago           musing_goldstine
5b45e476751c   ubuntu   "/bin/bash"   14 minutes ago   Exited (0) 11 minutes ago           laughing_faraday
57c8a4e32a69   ubuntu   "/bin/bash"   18 minutes ago   Exited (0) 14 minutes ago           busy_kowalevski
1cff14aff268   redis    "docker-entrypoint.s..." 2 months ago     Exited (0) 2 months ago           charming_gould
8acc94f1e645   hello-world "/hello"     2 months ago     Exited (0) 2 months ago           hungry_leavitt
root@anonymous:~# docker run -it -v volume1:/mnt ubuntu
root@54814eaf995b:/# cd mnt
root@54814eaf995b:/mnt# ls
test2.txt test3.txt
root@54814eaf995b:/mnt# exit
exit
root@anonymous:~# docker run -it ubuntu
root@3785c07ff637:/# cd mnt
root@3785c07ff637:/mnt# ls
root@3785c07ff637:/mnt# exit
exit
root@anonymous:~# docker volume ls
DRIVER   VOLUME NAME
local    7d5d77b9455f11d1ef3b20c4ec72fa3d07dac576ab78933607b21036a07cd74a
local    volume1
```

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.

A terminal window titled 'Terminal' with a dark background. The top bar shows 'Applications', 'Places', and 'Terminal' tabs. The status bar at the top right indicates 'Mon Feb 1 8:16 PM', a battery level of 70%, and a window count of 1. The terminal content shows a root user at an 'anonymous' host. The first command is 'docker volume inspect volume1', which outputs a JSON object containing details about the volume, including its name, driver, and mountpoint. The second command is 'ls /var/lib/docker/volumes/volume1/_data', which lists two files: 'test2.txt' and 'test3.txt'.

```
root@anonymous:~# docker volume inspect volume1
[
  {
    "CreatedAt": "2021-02-01T20:12:59+05:30",
    "Driver": "local",
    "Labels": null,
    "Mountpoint": "/var/lib/docker/volumes/volume1/_data",
    "Name": "volume1",
    "Options": null,
    "Scope": "local"
  }
]
root@anonymous:~# ls /var/lib/docker/volumes/volume1/_data
test2.txt test3.txt
root@anonymous:~#
```

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 use volume as the file sharing between different containers**

eg: cd root
touch test4.txt
ls

eg: cd mnt
ls

ApplicationsPlacesTerminal

Mon Feb 1 8:23 PM

1

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

~ #

Plain TextTab Width: 8Ln 1, Col 1INS

ApplicationsPlacesTerminal

Mon Feb 1 8:22 PM

1

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#

Plain TextTab Width: 8Ln 1, Col 1INS

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

docker volume rm [volume-name]