

## Assignment

1. Install Docker, either on your native OS or on a VM. Make sure it runs. type "docker -v" to check if it's installed.

try below commands for help

docker --help ---> This command shows all available options and commands to work with images and containers

docker images --help ---> This command shows all the available options and commands to work with docker images

docker ps --help ---> This command shows all the available options and commands to work with docker containers

Ans: Docker version 20.10.6, build 370c289

2. Find a image from dockerhub of your choice(recommended: nginx), don't use browser, pull the official image from dockerhub

Ans: docker pull nginx

3. List all the available images in your machine/vm, make sure you see recently pulled image in the list.

Ans: docker images

4. Find out the "Full" ImageId of the image that you pulled and write it below.

Ans: docker images --no-trunc

>>> sha256:f0b8a9a541369db503ff3b9d4fa6de561b300f7363920c2bff4577c6c24c5cf6

5. Create a container of your image

Ans: docker run nginx

6. List all the running containers

Ans: docker ps

7. List all the running and stopped containers

Ans docker ps -aq

8. Find out the "Full" containerId of the container and write it below.

Ans: docker ps --no-trunc

>>> 8c1b9fc00c3555d13df7012dd31e7d4ac048f9ad46e03ea460659cf33a8c415d

9. Find out how many image layers are used to build this image.

Ans: docker history nginx

Or docker inspect <image-id>

10. Get the Apache Tomcat 7 server image from the docker hub.

Ans: docker pull tomcat:7.0

11. Run the Apache Tomcat 7, I mean create a container of Apache Tomcat.

Ans: docker run tomcat:7.0

12. Find out what is the IP Address of the Apache Tomcat Container that it is running on

Ans: docker inspect --format '{{.NetworkSettings.IPAddress}}' ef945b18cf08

>>>'172.17.0.3'

13. Which Port it is using?

Ans: docker inspect --format '{{.NetworkSettings.Ports}}' ef945b18cf08

>>> 'map[8080/tcp:[]]'

14. Try to access the Tomcat's home page from your machine/vm.

Ans:

15. What is the disk size of Apache Tomcat image?

Ans: docker system df -v

or

docker inspect --format '{{.Size}}' e614000ce544 (put image ID)

16. Find out list of all environment variables that is configured for tomcat image, can you see JAVA\_HOME and CATALINA\_HOME? What did you notice about it?

Ans: docker run --env list tomcat:7.0

INFO: Java Home: /usr/local/openjdk-8/jre

INFO: CATALINA\_HOME: /usr/local/tomcat

or

docker exec ef945b18cf08 env

```
C:\Users\hp>docker exec ef945b18cf08 env
PATH=/usr/local/tomcat/bin:/usr/local/openjdk-8/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
HOSTNAME=ef945b18cf08
JAVA_HOME=/usr/local/openjdk-8
LANG=C.UTF-8
JAVA_VERSION=8u292
CATALINA_HOME=/usr/local/tomcat
TOMCAT_NATIVE_JIBDIR=/usr/local/tomcat/native-jni-lib
LD_LIBRARY_PATH=/usr/local/tomcat/native-jni-lib
GPG_KEYS=05A83110949707C93A279E3D3E6086867BA6 07E48665A34DCAF5E22E5E6266191C37C037D42 47309207D818FDD8DCD3F83F1931D68A307A10A5 541FBE7D8F78B25E05D0EE13C370389288584E7 61B832AC2F1C5A90F0F9B00A1C506407564C17
A3 713DA88BE5091153FE716F520880AB1D63011C7 79F7026C690BA50892CD8B66A3AD3F4F22CAFED 9BA44C2621385C966EBA586F72C284D731FABEE A276772899860B50844682F8ACB77FC2E86E29AC A9C5DF4D22E99998D9875A5110C01C5A2F6059E7 DCF
D35E0BFBCA7342752DE8B6F821E8933C60243 F3A04C59508586A5F1ECA43E3B7BB1000811BBE F7DA488B64BC8BA7EE6935CD23C100498E23
TOMCAT_MAJOR=7
TOMCAT_VERSION=7.0.109
TOMCAT_SHA512=612e830913bf1401bc9540e2273e465b0ee7ef63750a9969a80f1e9da9edb4888aa621fcc6fa5ba23cff94a40e91eb97e3f969b8064dab49b2d0ea29e59b57e
HOME=/root
```

17. Find out which port is exposed for tomcat?

Ans: docker port 7a16597526cd

>>> 8080/tcp -> 0.0.0.0:8383

18. Run multiple containers of tomcat on different port and access it's home page.

Ans: docker run -d -p 8888:8080 tomcat:7.0

docker run -d -p 8383:8080 tomcat:7.0

.....  
19. Pull ubuntu os from dockerhub, try to pull 2 images of ubuntu, Except the latest one.

Ans: docker pull ubuntu:xenial

docker pull ubuntu:rolling  
.....

20. Run the container of ubuntu in attached mode.

Ans: docker run -it ubuntu:xenial  
.....

21. Run the container of another ubuntu in detached mode.

Ans: docker run -d ubuntu:rolling  
.....

22. Check how many ubuntu containers are running and stopped

Ans: docker ps  
.....

23. Is the tomcat container running? If no, start one.

Ans: yes  
.....

24. Check the logs, generated by tomcat container(don't forget to make request to tomcat's home page to see the log).

Ans: docker logs 7a16597526cd  
.....

25. Check if the Ubuntu container is running? If no, start one in attached mode to the terminal.

Ans: docker ps

docker attach 9ff95a467e45  
.....

26. Login as root user in ubuntu container

Ans:

```
C:\Users\hp>docker run -it 9ff95a467e45
root@0c0ac3986a8e:/# whoami
root
root@0c0ac3986a8e:/#
```

.....

27. Create a file with any name in root directory

Ans:

```
C:\Users\hp>docker run -it 9ff95a467e45
root@0c0ac3986a8e:/# whoami
root
root@0c0ac3986a8e:/# touch myfile
root@0c0ac3986a8e:/# ls
bin boot dev etc home lib lib64 media mnt myfile opt proc root run sbin srv sys tmp usr var
root@0c0ac3986a8e:/#
```

.....

28. Install software of your choice in ubuntu container using "apt-get install"

Ans:

```
root@0c0ac3986a8e:/# ls
bin boot dev etc home lib lib64 media mnt myfile opt proc root run sbin srv sys usr var
root@0c0ac3986a8e:/# apt-get update
Get:1 http://archive.ubuntu.com/ubuntu xenial InRelease [247 kB]
Get:2 http://security.ubuntu.com/ubuntu xenial-security InRelease [109 kB]
Get:3 http://archive.ubuntu.com/ubuntu xenial-updates InRelease [109 kB]
Get:4 http://archive.ubuntu.com/ubuntu xenial-backports InRelease [107 kB]
Get:5 http://archive.ubuntu.com/ubuntu xenial/main amd64 Packages [1558 kB]
Get:6 http://security.ubuntu.com/ubuntu xenial-security/main amd64 Packages [2051 kB]
Get:7 http://archive.ubuntu.com/ubuntu xenial/restricted amd64 Packages [14.1 kB]
Get:8 http://archive.ubuntu.com/ubuntu xenial/universe amd64 Packages [9827 kB]
Get:9 http://security.ubuntu.com/ubuntu xenial-security/restricted amd64 Packages [15.9 kB]
Get:10 http://security.ubuntu.com/ubuntu xenial-security/universe amd64 Packages [984 kB]
Get:11 http://security.ubuntu.com/ubuntu xenial-security/multiverse amd64 Packages [8820 B]
Get:12 http://archive.ubuntu.com/ubuntu xenial/multiverse amd64 Packages [176 kB]
Get:13 http://archive.ubuntu.com/ubuntu xenial-updates/main amd64 Packages [2559 kB]
Get:14 http://archive.ubuntu.com/ubuntu xenial-updates/restricted amd64 Packages [16.4 kB]
Get:15 http://archive.ubuntu.com/ubuntu xenial-updates/universe amd64 Packages [1544 kB]
Get:16 http://archive.ubuntu.com/ubuntu xenial-updates/multiverse amd64 Packages [26.2 kB]
Get:17 http://archive.ubuntu.com/ubuntu xenial-backports/main amd64 Packages [10.9 kB]Get:18 http://archive.ubuntu.com/ubuntu xenial-backports/universe amd64 Packages [12.7 kB]
Fetched 19.4 MB in 58s (333 kB/s)
Reading package lists... Done
root@0c0ac3986a8e:/# apt-get install tree
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  tree
0 upgraded, 1 newly installed, 0 to remove and 5 not upgraded.
Need to get 40.6 kB of archives.
After this operation, 138 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu xenial/universe amd64 tree amd64 1.7.0-3 [40.6 kB]
Fetched 40.6 kB in 1s (33.4 kB/s)
debconf: delaying package configuration, since apt-utils is not installed
Selecting previously unselected package tree.
(Reading database ... 4785 files and directories currently installed.)
Preparing to unpack .../tree_1.7.0-3_amd64.deb ...
Unpacking tree (1.7.0-3) ...
Setting up tree (1.7.0-3) ...
root@0c0ac3986a8e:/#
```

29. Now exit the ubuntu shell, are you back to your host machine, if not, come back to the host machine.

```
root@0c0ac3986a8e:/# apt-get install tree
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  tree
0 upgraded, 1 newly installed, 0 to remove and 5 not upgraded.
Need to get 40.6 kB of archives.
After this operation, 138 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu xenial/universe amd64 tree amd64 1.7.0-3 [40.6
kB]
Fetched 40.6 kB in 1s (33.4 kB/s)
debconf: delaying package configuration, since apt-utils is not installed
Selecting previously unselected package tree.
(Reading database ... 4785 files and directories currently installed.)
Preparing to unpack .../tree_1.7.0-3_amd64.deb ...
Unpacking tree (1.7.0-3) ...
Setting up tree (1.7.0-3) ...
root@0c0ac3986a8e:/# exit
exit
C:\Users\hp>
```

Yes,back to host machine

30. Check if the ubuntu container is running.

Ans. Yes container is running

31. Create a new ubuntu container out of the same image as that previous container in attached mode.

Ans. docker run -it ubuntu:xenial

32. Login as a root user

```
C:\Users\hp>docker run -it ubuntu:xenial
root@4ef2604d066b:/# whoami
root
root@4ef2604d066b:/#
```

Ans:

33. Check if you can see the file created in previous container, you will not see the file as well as software that you installed in the previous container. Now kill this Container.

Ans:

```
C:\Users\hp>docker run -it ubuntu:xenial
root@4ef2604d066b:/# whoami
root
root@4ef2604d066b:/# ls
bin  dev  home  lib64  mnt  proc  run  srv  tmp  var
boot  etc  lib  media  opt  root  sbin  sys  usr
root@4ef2604d066b:/# exit
exit
```

docker kill 4ef2604d066b (container ID)

34. Do you have the previous ubuntu container where you created the file and installed the software? If no repeat step 25 to 29.

Ans: I have previous ubuntu container

```
C:\Users\hp>docker attach 0c0ac3986a8e
root@0c0ac3986a8e:/#
root@0c0ac3986a8e:/# ls
bin  dev  home  lib64  mnt  opt  root  sbin  sys  usr
boot  etc  lib  media  myfile  proc  run  srv  tmp  var
root@0c0ac3986a8e:/# which tree
/usr/bin/tree
root@0c0ac3986a8e:/#
```

35. Create an Image out of the existing container.

Ans:

```
C:\Users\hp>docker commit 0c0ac3986a8e prachi
sha256:3bf8424dbcd8cfef64c8fe9a8eeff6221a43f5b855e2fe9789597e40dd23af
```

36. Now Create a Container out of this image and login into it to see if you can see the file and software installed by you in the previous container.

Ans: yes, have software installed in the previous container

```
C:\Users\hp>docker commit 0c0ac3986a8e prachi
sha256:3bf8424dbcd8cfef64c8fe9a8eeff6221a43f5b855e2fe9789597e40dd23af

C:\Users\hp>docker run -it prachi
root@53f79d349a26:/# ls
bin boot dev etc home lib lib64 media mnt myfile opt proc root run sbin srv sys tmp usr var
root@53f79d349a26:/# which tree
/usr/bin/tree
root@53f79d349a26:/#
```

37. Do you have running tomcat container? If yes, Stop it and kill all tomcat container.

Ans: docker stop 39b0038b31be 7a16597526cd

```
C:\Users\hp>docker ps
```

CONTAINER ID	IMAGE	NAMES	COMMAND	CREATED	STATUS	PORTS
4ef2604d066b	ubuntu:xenial	quizzical_hofstadter	"/bin/bash"	48 minutes ago	Up 36 minutes	
6c2728dae982	docker/getting-started	elated_mclaren	"/docker-entrypoint..."	16 hours ago	Up 6 seconds	0.0.0.0:80->80/tcp, :::80->80/tcp

38. Create an index.html file with following code in it:-

```
<h1>This is Tomcat Container</h1>
```

Now, Start a tomcat container in such a way that on hitting its URL for home page it should show the above html page.

```
C:\Users\hp>docker exec -it 27295c167583 bash
root@27295c167583:/usr/local/tomcat# ls
LICENSE NOTICE RELEASE-NOTES RUNNING.txt bin conf include lib logs native-jni-lib temp webapps work
root@27295c167583:/usr/local/tomcat# cd webapps
root@27295c167583:/usr/local/tomcat/webapps# cd ROOT
root@27295c167583:/usr/local/tomcat/webapps/ROOT# echo "<h1>this is tomcat Container</h1>" > index.html
root@27295c167583:/usr/local/tomcat/webapps/ROOT# cat index.html
<h1>this is tomcat Container</h1>
root@27295c167583:/usr/local/tomcat/webapps/ROOT#
```

At port localhost:9787 this will appear



39. type below command:-

`docker images --help`

Now, try to run command that proves the concept of following three options:-

1. -a
2. -f
3. -q

```
Command Prompt

C:\Users\hp>docker images -a
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
prachi              latest             3bf8424dbcdf       34 minutes ago     167MB
<none>              <none>             71941adaa9d3       42 minutes ago     135MB
ubuntu              xenial             9ff95a467e45       3 days ago         135MB
docker/getting-started latest             3ba8f2ff0727       2 months ago       27.9MB
tomcat              8.0                ef6a7c98d192       2 years ago        356MB

C:\Users\hp>docker images -q
3bf8424dbcdf
71941adaa9d3
9ff95a467e45
3ba8f2ff0727
ef6a7c98d192

C:\Users\hp>docker images list --quiet --filter dangling=true
```

40. type below command:-

`docker ps --help`

Now, try to run command that proves the concept of following six options:-

1. -a
2. -f
3. -q
4. -n
5. -l
6. -s

```
C:\Users\hp>docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
597a270a5dd7   ef6a7c98d192   "catalina.sh run"       43 minutes ago Exited (130) 31 minutes ago      0.0.0.0:9787->8080/tcp, :::9787->8080/tcp   compassionate_robinson
27295c167583   tomcat:8.0     "catalina.sh run"       49 minutes ago Up 49 minutes                               competent_neumann
63f70d349a26   prachi        "/bin/bash"            About an hour ago Exited (0) 55 minutes ago                               confident_hermann
c1b7909bd553f   71941adaa9d3   "/bin/bash"            About an hour ago Exited (0) About an hour ago                               lucid_turing
4ef20d4d066b   ubuntu:xenial  "/bin/bash"            2 hours ago     Up 2 hours                               quizzical_hofstadter
c0ad6f1172b1   9ff95a467e45   "/bin/bash"            2 hours ago     Exited (0) 19 minutes ago                               gallant_grothendieck
0c0ac3986a8e   9ff95a467e45   "/bin/bash"            2 hours ago     Exited (0) About an hour ago                               competent_lovelace
0c2728dae982   docker/getting-started "/docker-entrypoint..." 17 hours ago    Up 55 minutes    0.0.0.0:80->80/tcp, :::80->80/tcp   elated_mclaren

C:\Users\hp>docker ps -f name=elated_mclaren
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
0c2728dae982   docker/getting-started "/docker-entrypoint..." 17 hours ago    Up 55 minutes    0.0.0.0:80->80/tcp, :::80->80/tcp   elated_mclaren

C:\Users\hp>docker ps -q
27295c167583
4ef20d4d066b
0c2728dae982

C:\Users\hp>docker ps -n 2
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
597a270a5dd7   ef6a7c98d192   "catalina.sh run"       44 minutes ago Exited (130) 31 minutes ago      0.0.0.0:9787->8080/tcp, :::9787->8080/tcp   compassionate_robinson
27295c167583   tomcat:8.0     "catalina.sh run"       49 minutes ago Up 49 minutes                               competent_neumann

C:\Users\hp>docker ps -l
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS        NAMES
597a270a5dd7   ef6a7c98d192   "catalina.sh run"       44 minutes ago Exited (130) 31 minutes ago      0.0.0.0:9787->8080/tcp, :::9787->8080/tcp   compassionate_robinson

C:\Users\hp>docker ps -s
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES        SIZE
27295c167583   tomcat:8.0     "catalina.sh run"       50 minutes ago Up 49 minutes    0.0.0.0:9787->8080/tcp, :::9787->8080/tcp   competent_neumann   39.6kB (virtual 356MB)
4ef20d4d066b   ubuntu:xenial  "/bin/bash"            2 hours ago     Up 2 hours       0.0.0.0:80->80/tcp, :::80->80/tcp           quizzical_hofstadter 15B (virtual 135MB)
0c2728dae982   docker/getting-started "/docker-entrypoint..." 17 hours ago    Up 55 minutes    0.0.0.0:80->80/tcp, :::80->80/tcp           elated_mclaren       1.12kB (virtual 27.9MB)

C:\Users\hp>
```

41. Type below command:-

```
docker --help
```

you will various sections of commands apart from options like "Management Commands" and "Commands".

Write some texts below describing the use of "Management Commands".

Use each command mentioned below and prove its concepts as described in the --help description. write what you have understood from the output of the command after its successful execution.

1. **cp** – It Copy files/folders between a container and the local filesystem
2. **create** – It Creates a new container but does not run it, we need to start it separately.
3. **export** - Export a container's filesystem as a tar archive, docker export <container-name> > example.tar



**4. history** - Shows the history of an image

**5. info** – It displays system-wide information

**6. login** – It is used to log in to a Docker registry

**7. logout** - It is used to log out from a Docker registry

**8. rename** - Renames a container

**9. save** - save one or more images to a tar archive,

`docker save <container-name> > example.tar`

**10. stats** – It displays a live stream of container(s) resource usage statistics

**11. top** - Display the running processes of a container

.....

42. Kill all running container in one liner command.

Ans: `docker container kill $(docker ps -q)`

or

`docker rm -f $(docker ps -qa)`

.....

43. Delete all images in one liner command.

Ans: `docker rmi $(docker images -a -q)`

Or

`Docker images -a -q | xargs docker rmi -f`

.....

44. Create a simple Dockerfile, build it and run it.

Ans:

```
Dockerfile - Notepad
File Edit Format View Help
# getting base image ubuntu
FROM ubuntu
MAINTAINER prachipatl05 <prachipatl1314@gmail.com>
RUN apt-get update
CMD ["echo", "hello world"]

Ln 3, Col 22    100%    Unix (LF)    UTF-8
```

```
C:\Users\hp\Desktop\Dockerfiles>docker build -t myimg1:1.0 .
[+] Building 134.5s (7/7) FINISHED
=> [internal] load build definition from Dockerfile 3.6s
=> => transferring dockerfile: 179B 0.1s
=> [internal] load .dockerignore 2.9s
=> => transferring context: 2B 0.2s
=> [internal] load metadata for docker.io/library/ubuntu:latest 25.4s
=> [auth] library/ubuntu:pull token for registry-1.docker.io 0.0s
=> [1/2] FROM docker.io/library/ubuntu@sha256:cf31af331f38d1d7158470e095b132acd126a7180a54f263d386da88eb681d93 50.6s
=> => resolve docker.io/library/ubuntu@sha256:cf31af331f38d1d7158470e095b132acd126a7180a54f263d386da88eb681d93 0.3s
=> => sha256:cf31af331f38d1d7158470e095b132acd126a7180a54f263d386da88eb681d93 1.20kB / 1.20kB 0.0s
=> => sha256:86ac87f73641c920fb42cc9612d4fb57b5626b56ea2a19b894d0673fd5b4f2e9 943B / 943B 0.0s
=> => sha256:7e0aa2d69a153215c790488ed1fcec162015e973e49962d438e18249d16fa9bd 3.31kB / 3.31kB 0.0s
=> => sha256:5e9250ddb7d0fa6d13302c7c3e6a0aa40390e42424caed1e5289077ee4054709 187B / 187B 14.4s
=> => sha256:345e3491a907bb7c6f1bdddcf4a94284b8b6ddd77eb7d93f09432b17b20f2bbe 28.54MB / 28.54MB 40.4s
=> => sha256:57671312ef6fdbcfc340e5fed0fb0863350cd806c92b1fdd7978adbd02afc5c3 851B / 851B 14.6s
=> => extracting sha256:345e3491a907bb7c6f1bdddcf4a94284b8b6ddd77eb7d93f09432b17b20f2bbe 2.0s
=> => extracting sha256:57671312ef6fdbcfc340e5fed0fb0863350cd806c92b1fdd7978adbd02afc5c3 0.0s
=> => extracting sha256:5e9250ddb7d0fa6d13302c7c3e6a0aa40390e42424caed1e5289077ee4054709 0.0s
=> [2/2] RUN apt-get update 50.9s
=> exporting to image 1.4s
=> => exporting layers 1.1s
=> => writing image sha256:d953de184e4bfda31acb053ca3aa57d2e58f82effde6e7798860c7b5945a232 0.1s
=> => naming to docker.io/library/myimg1:1.0 0.1s

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
```

```
C:\Users\hp\Desktop\Dockerfiles>docker run d953de184e4b
hello world
```

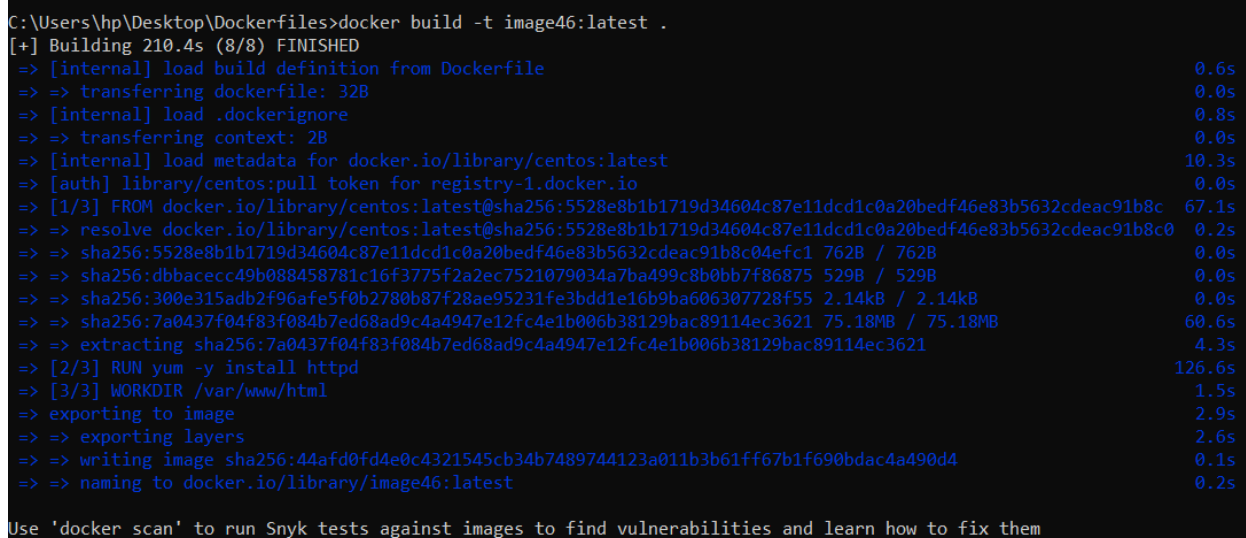
.....

45. Create one or more Dockerfile that demonstrate the following commands in Dockerfile (Write a PoC for each in one or more Dockerfile)

1. USER
2. RUN
3. ENV
4. CMD
5. RUN
6. ENTRYPOINT
7. EXPOSE
8. VOLUME



```
1 FROM centos:latest
2 MAINTAINER prachipatl05
3 ENV WORKDIR /var/www/html
4 RUN yum -y install httpd
5 VOLUME C:\Users\hp\Desktop\ppp
6 USER root
7 WORKDIR $workdir
8 CMD ["/usr/sbin/httpd", "-D", "FOREGROUND"]
9 EXPOSE 8383
10 ENTRYPOINT ["centos"]
```



```
C:\Users\hp\Desktop\Dockerfiles>docker build -t image46:latest .
[+] Building 210.4s (8/8) FINISHED
=> [internal] load build definition from Dockerfile                                0.6s
=> => transferring dockerfile: 32B                                                0.0s
=> [internal] load .dockerignore                                                  0.8s
=> => transferring context: 2B                                                    0.0s
=> [internal] load metadata for docker.io/library/centos:latest                 10.3s
=> [auth] library/centos:pull token for registry-1.docker.io                   0.0s
=> [1/3] FROM docker.io/library/centos:latest@sha256:5528e8b1b1719d34604c87e11dcd1c0a20bedf46e83b5632cdeac91b8c 67.1s
=> => resolve docker.io/library/centos:latest@sha256:5528e8b1b1719d34604c87e11dcd1c0a20bedf46e83b5632cdeac91b8c0 0.2s
=> => sha256:5528e8b1b1719d34604c87e11dcd1c0a20bedf46e83b5632cdeac91b8c04efc1 762B / 762B 0.0s
=> => sha256:dbbacecc49b088458781c16f3775f2a2ec7521079034a7ba499c8b0bb7f86875 529B / 529B 0.0s
=> => sha256:300e315adb2f96afe5f0b2780b87f28ae95231fe3bdd1e16b9ba606307728f55 2.14kB / 2.14kB 0.0s
=> => sha256:7a0437f04f83f084b7ed68ad9c4a4947e12fc4e1b006b38129bac89114ec3621 75.18MB / 75.18MB 60.6s
=> => extracting sha256:7a0437f04f83f084b7ed68ad9c4a4947e12fc4e1b006b38129bac89114ec3621 4.3s
=> [2/3] RUN yum -y install httpd                                              126.6s
=> [3/3] WORKDIR /var/www/html                                                1.5s
=> exporting to image                                                         2.9s
=> => exporting layers                                                            2.6s
=> => writing image sha256:44afd0fd4e0c4321545cb34b7489744123a011b3b61ff67b1f690bdac4a490d4 0.1s
=> => naming to docker.io/library/image46:latest                               0.2s

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
```

.....

#### 46. Dockerhub:-

Find a application you care about on docker hub.

Launch the container.

Install another application in it.

Save (commit) the image.

Upload that to docker hub in your account and share it with a colleague, ask them to use your image and run the container out of it in their machine/vm.

Ans:

Converted the container with ubuntu+nginx installed to docker image  
and pushed it docker hub

```
C:\Users\hp>docker push prachipatle05/ubuntu-nginx45
Using default tag: latest
The push refers to repository [docker.io/prachipatle05/ubuntu-nginx45]
1fe7f2dbb80e: Pushed
cdc4d0cd3d15: Mounted from library/ubuntu
43d05a2c9517: Mounted from library/ubuntu
1902cf049561: Pushed
9fe04650404a: Mounted from library/ubuntu
latest: digest: sha256:02afafd4064cb627074be54305cbeac7bac87cfc27bb57f6fd075b5aefef44ad size: 1362
```

Others can use my image from--

<https://hub.docker.com/repository/docker/prachipatle05/ubuntu-nginx45>