Assignment

NOTE: Do not forget to see manual page using "--help" option in command when searching for options/commands for a particular task.

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 avaialble options and commands to work with docker images

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

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

**Ans**: docker pull nginx:latest

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 --digests

5. Create a container of your image

**Ans**: docker run -it hello-world

6. List all the running containers

**Ans**: docker ps

7. List all the running and stopped containers

**Ans**: docker ps -a

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

**Ans**: docker inspect --format="{{.Id}}" dreamy\_swirles

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

Ans: docker inspect <image name>

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

**Ans**: docker pull tomcat

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

**Ans**: docker run -d tomcat

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

Ans: docker inspect sharp\_panini | grep '"IPAddress"' | head -n 1

13. Which Port it is using?

**Ans**: docker ps

Port: 8080

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

**Ans**: curl http://172.17.0.2:8080

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

Ans: 438 MB

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 exec -it <container name> /bin/bash

env echo $CATALINA\_HOME

env echo $JAVA\_HOME

17. Find out which port is exposed for tomcat?

**Ans**: docker inspect <container name>

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

**Ans**:

docker run -d tomcat

docker run -d -p8089 tomcat:latest

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

**Ans**:

docker pull ubuntu-debootstrap

docker pull ubuntu

20. Run the container of ubuntu in attached mode.

**Ans**: docker attach <container name >

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

**Ans**: docker run -d ubuntu-debootstrap:latest

22. Check how many ubuntu containers are running and stopped

Ans: docker ps -a --filter ancestor=ubuntu

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

**Ans**: docker start <container id>

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 <container id>

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

**Ans**: docker run -it <container id>

26. Login as root user in ubuntu container

**Ans**: docker run -u 0 -it ubuntu /bin/bash

27. Create a file with any name in root directory

**Ans**: touch a.txt

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

Ans: apt-get install -y docker.io

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

**Ans**: exit

30. Check if the ubuntu container is running.

**Ans**: No

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

**Ans**: docker run -u 0 -it ubuntu /bin/bash

32. Login as a root user

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: File is not there

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

Ans:

35. Create an Image out of the existing container.

**Ans**: docker commit de1691cf4ce9 t1\_image

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**: docker run -u 0 -it t1\_image /bin/bash

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

**Ans**: docker rm $(docker ps -a -q)

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

<h1>This is Tomcat Container</h1>

vi index.html

chmod u+x index.html

chmod g+x index.html

chmod o+x index.html

chmod u+w index.html

chmod g+w index.html

chmod o+w index.html

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

39. type below command:-

docker images --help

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

1. -a

Ans: docker images -a

docker images -all

2. -f

docker images -f

3. -q

docker images -q

write atleast 1 command using each option above and prove their concepts as described in the --help.

40. type below command:-

docker ps --help

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

1. -a

docker ps -a

2. -f

docker ps -f

3. -q

docker ps -q

4. -n

docker ps -n1 – last created container

5. -l

docker ps -l – latest created container

6. -s

docker ps -s – file size of the container

write atleast 1 command using each option above and prove their concepts as described in the --help.

41. Type below command:-

docker --help

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

config Manage Docker configs

container Manage containers

image Manage images

network Manage networks

node Manage Swarm nodes

plugin Manage plugins

secret Manage Docker secrets

service Manage services

stack Manage Docker stacks

swarm Manage Swarm

system Manage Docker

trust Manage trust on Docker images

volume Manage volumes

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

Management command is used for checking the images pulled from docker hub.

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

**Ans**: docker cp ‘priceless\_knut’ ‘source path’ ‘destination path’

2. create

**Ans**: docker create hello-world

3. export

**Ans**: docker export priceless\_knuth

4. history

Ans: docker history hello-world

5. info

Ans: docker info – get the all info on the container

6. login

Ans: docker login – login using docker hub credentials to push and pull images

7. logout

Ans: docker logout – to logout from the above command

8. rename

Ans: docker rename priceless\_knuth ashwath

9. save

10. stats

Ans: docker stats – to get the statics of the container running

11. top

Ans : docker top <container id>

42. Kill all running container in one liner command.

Ans: docker rm $(docker ps -a -q)

43. Delete all images in one liner command.

Ans: docker rmi $(docker images -q)

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

Ans:

Vi Dockerfile

FROM ubuntu

RUN /bin/bash

docker build --tag ash .

docker images

45. Create one or more Dockerfile that demostrate 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

Ans:

FROM ubuntu:latest

LABEL maintainer="myname@somecompany.com"

#USER ashwashe

RUN apt-get update && apt-get upgrade -y

RUN apt-get install nginx -y

EXPOSE 80

CMD ["nginx", "-g", "daemon off;"]

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.

docker run -u 0 -it ubuntu /bin/bash

docker commit 8e7fe92eabd5 testimage

export DOCKER\_ID\_USER="ashwashe"

docker login

--login using docker hub credentials

docker tag testimage $DOCKER\_ID\_USER/testimage

docker push $DOCKER\_ID\_USER/testimage

URL : https://hub.docker.com/r/ashwashe/testimage/