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.

$ Docker version 19.03.1, build 74b1e89e8a

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

$ docker pull nginx

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

$ docker images -a

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

$ docker images nginx -q

```fa5269854a5e```

5. Create a container of your image

$ docker run --detach --publish 80:80 --name webserver nginx

6. List all the running containers

$ docker ps

7. List all the running and stopped containers

$ docker ps -a

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

$ docker ps -a -q

```b8f8f8b8b8b8```

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

$ docker history nginx

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

$ docker pull tomcat:7

11. Run the Apache Tomcat 7

$ docker run -d -p 8080:8080 --name tomcat7 tomcat:7

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

$ docker inspect tomcat7

or

$ docker inspect -f "{{ .NetworkSettings.IPAdress }}" tomcat7

```172.17.0.2```

13. Which Port it is using?

$ docker inspect tomcat7

`"ExposedPorts": {

"8080/tcp" `

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

http://host-ip:port/

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

$ docker system df -v

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?

$docker inspect -f "{{ .Config.Env }}" tomcat7

or

$ docker inspect tomcat7

> Yes , I see JAVA\_HOME and CATALINA\_HOME.

> - The JAVA\_HOME is pointing to the java installation directory.

> - The CATALINA\_HOME is pointing to the tomcat installation directory.

17. Find out which port is exposed for tomcat?

$ docker inspect -f "{{ .HostConfig }}" tomcat7

or

$ docker inspect tomcat7

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

Running 1st container on port 8080

$ docker run -d -p 8080:8080 --name tomcat7 tomcat:7

Running 2nd container on port 8081

$ docker run -d -p 8081:8080 --name tomcat7ii tomcat:7

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

$ docker pull ubuntu:16.04

$ docker pull ubuntu:14.04

$ docker images

20. Run the container of ubuntu in attached mode.

$ docker run -it --name ubu16 ubuntu:16.04

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

$ docker run -d --name ubu14 ubuntu:14.04

22. Check how many ubuntu containers are running and stopped

$ docker ps

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

$ docker ps

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

$ docker logs tomcat7

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

$ docker ps

26. Login as root user in ubuntu container

$ docker exec -it ubu16 /bin/bash

27. Create a file with any name in root directory

# touch file.txt

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

# apt-get update

# apt-get install rolldice

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

$ exit

30. Check if the ubuntu container is running.

$ docker ps

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

$ docker run -it --name ubu16\_1 ubuntu:16.04

32. Login as a root user

$ docker exec -it ubu16\_1 /bin/bash

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.

# exit

$ docker kill ubu16\_1

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

$docker exec -it ubu16 /bin/bash

35. Create an Image out of the existing container.

$ docker commit ubu16 ubu16\_2

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.

$ docker run -it --name ubu16\_2 ubu16\_2

# ls

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

$ docker ps

$ docker stop tomcat7

$ docker stop tomcat7ii

$ docker ps

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.

$ docker run -d -p 8080:8080 --name tomcat7 tomcat:7

$ docker ps

39. Now, access the tomcat container's home page.

$ docker exec -it tomcat7 /bin/bash

$ curl http://host-ip:port/

39. type below command:-

docker images --help

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

> - `-a` - Show all images

> - ` -f` - Filter output based on conditions provided

> - ` -q ` - Show only the numeric image IDs

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

$ docker images --help

$ docker images -a

$ docker images -f "dangling=true"

$ docker images -q

40. type below command:-

docker ps --help

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

> - -a - Show all containers

> - -f - Filter output based on conditions provided

> - -q - Show only the container IDs

> - -n - Show n last created containers (includes all

states) (default -1)

> - -l - Show the latest created container (includes all

states)

> - -s - Display total file sizes