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.

If you can't install or configure Docker, you can use the online docker setup to do the assignment.

Step1 Goto:- https://www.katacoda.com/courses/kubernetes/playground

Step2 Click on "continue" button on the left panel

Step3 Click on "launch.sh" button on the left panel

Step4 From the right panel use the top console to execute below command:-

docker -v

Try below commands for help

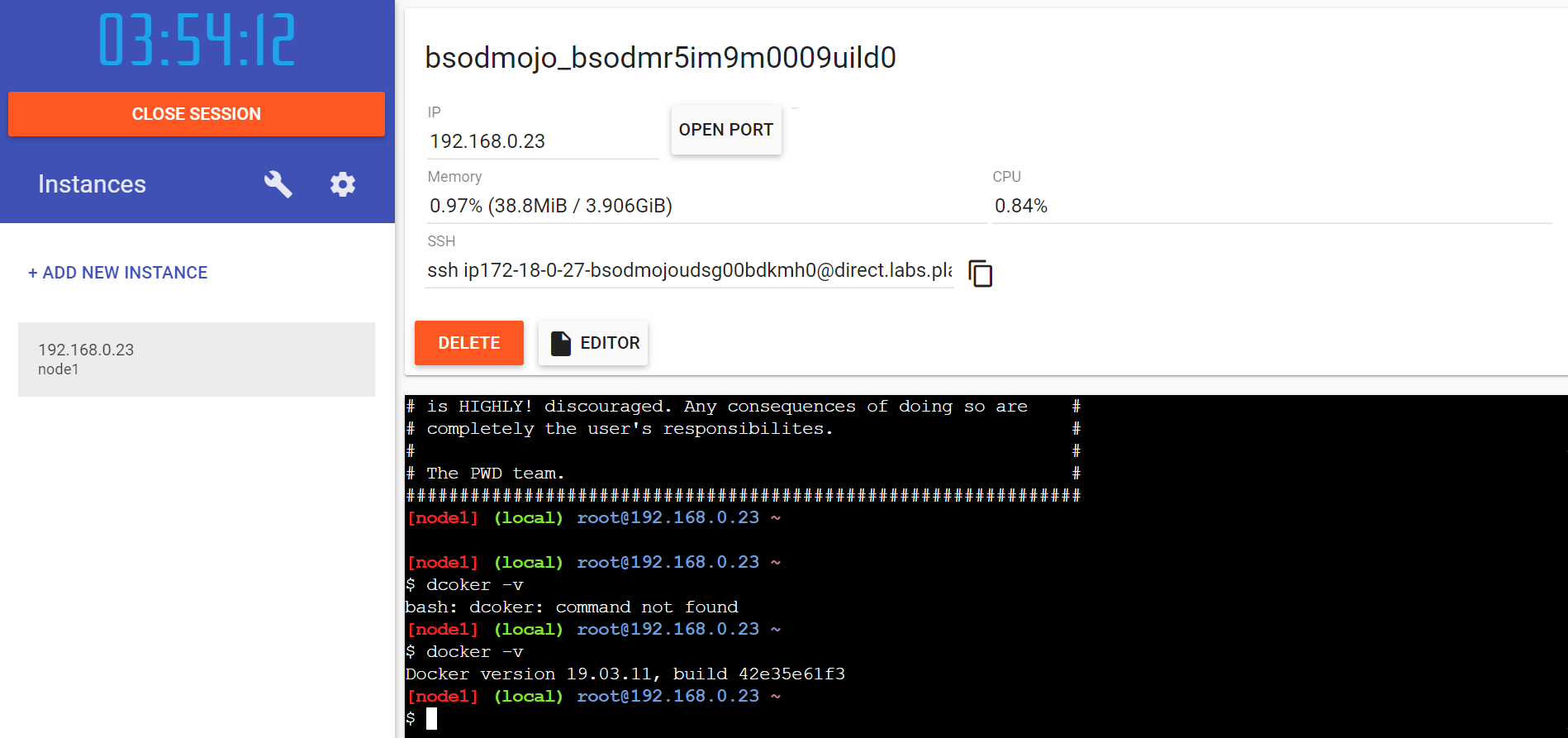
docker --help ---> This command shows all available options and commands to work with images and containers

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

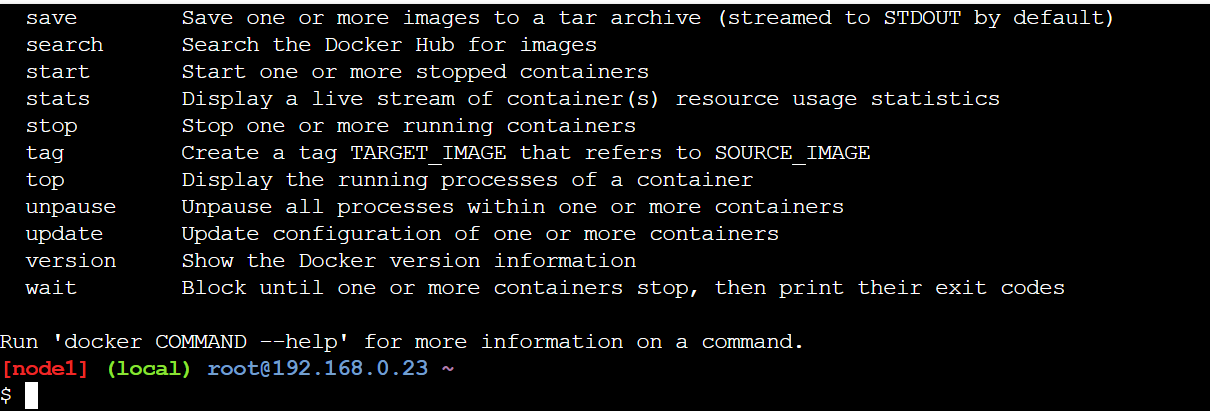
docker container --help ---> This command shows all the avaialble options and commands to work with docker containers

NOTE:- DO NOT TRY TO USE INTERNET TO SOLVE ASSIGNMENT, BETTER USE THE ABOVE --help OPTION TO SEE THE MANUAL OF ANY PARTICULAR COMMAND AND FIGURE OUT THE SOLUTIONS ON YOUR OWN.

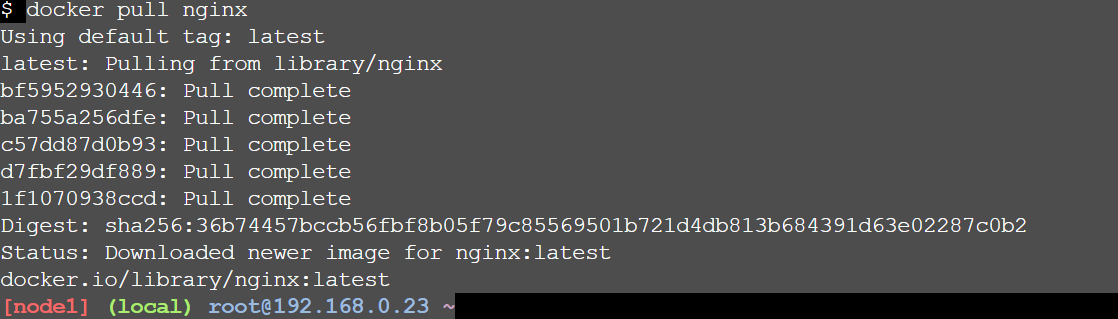
Ans: Docker playground using online:



docker –help

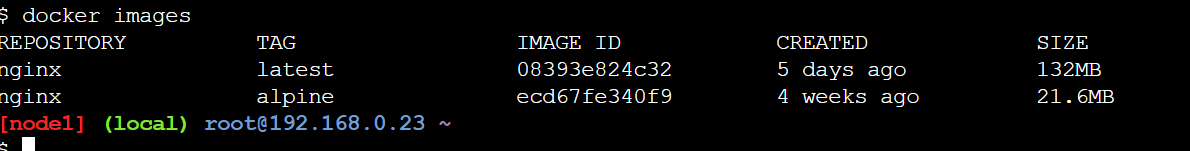


2. Find a image from dockerhub of your choice(recommeded: nginx), don't use browser, pull the official image from dockerhub

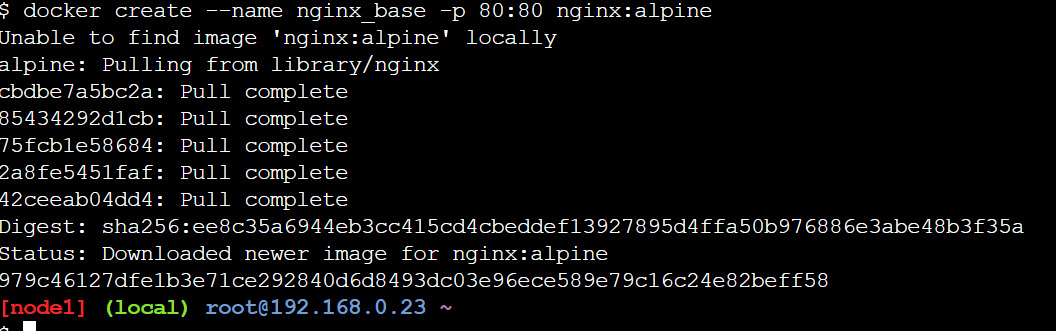
Ans:

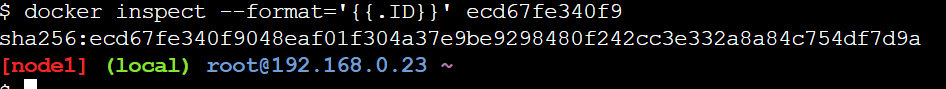
3. List all the available images in your machine/vm, make sure you see recently pulled image in the list.



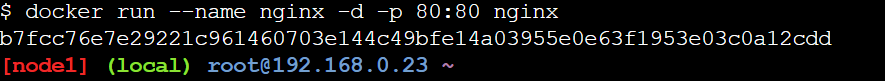


4. Find out the "Full" ImageId of the image that you pulled and write it below.

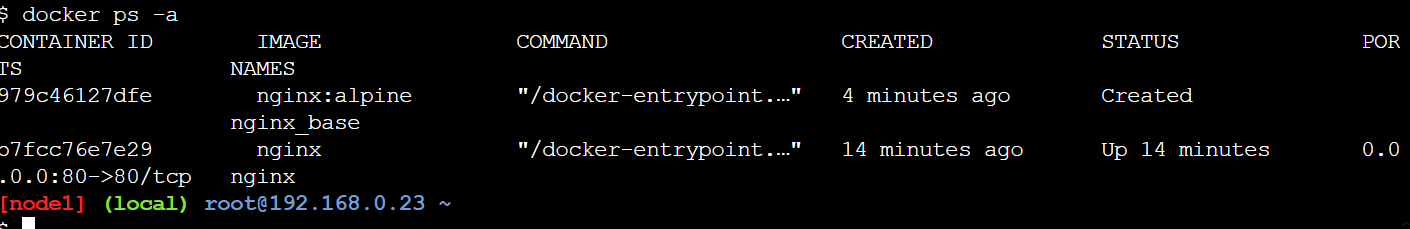




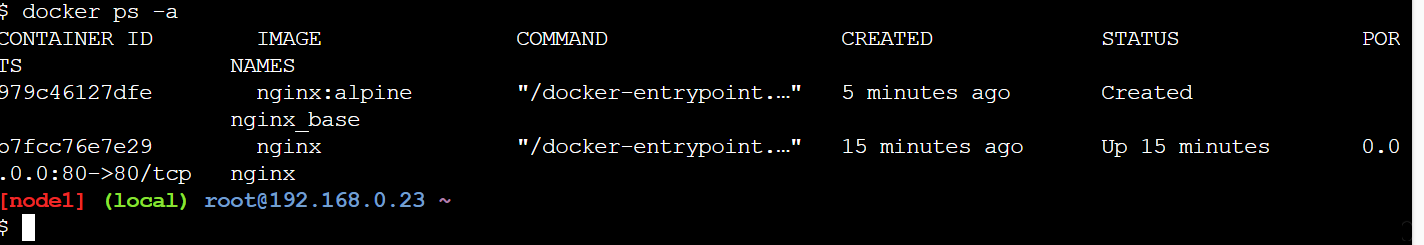
5. Create a container of your image



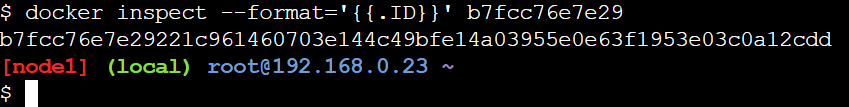
6. List all the running containers



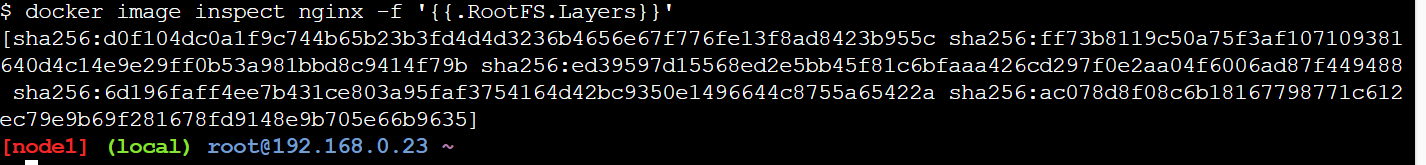
7. List all the running and stopped containers



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

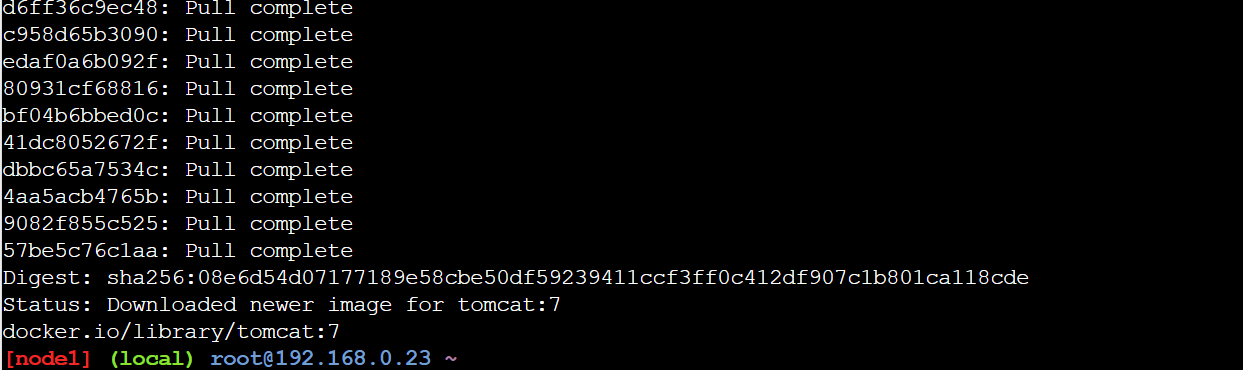


9. Find out how many image layers are used to build this image.

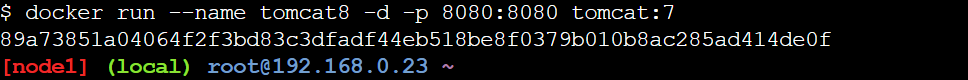


10. Get the Apache Tomcat 7 server image from the docker hub.

docker pull tomcat:7



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



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

docker inspect tomact8 |grep "IPAddress"

13. Which Port it is using?

docker port tomcat 7

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

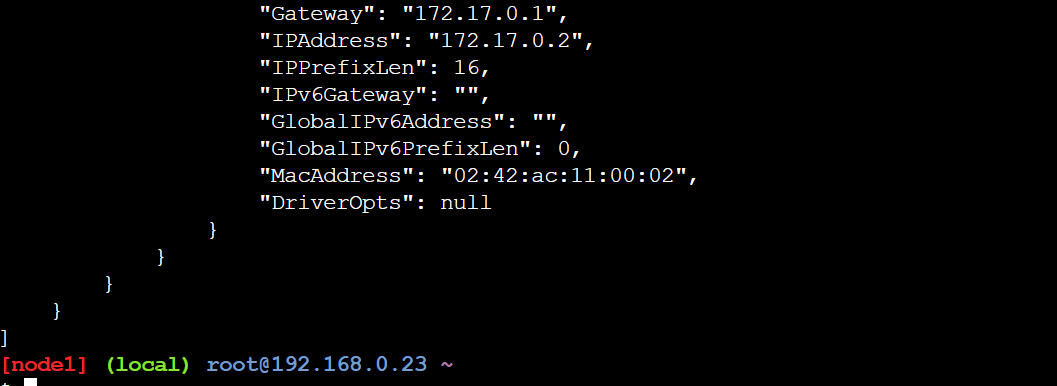
https://ipaddress:8080

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

docker image inspect --format ='{{.Size}}' tomcat:latest

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 <id>



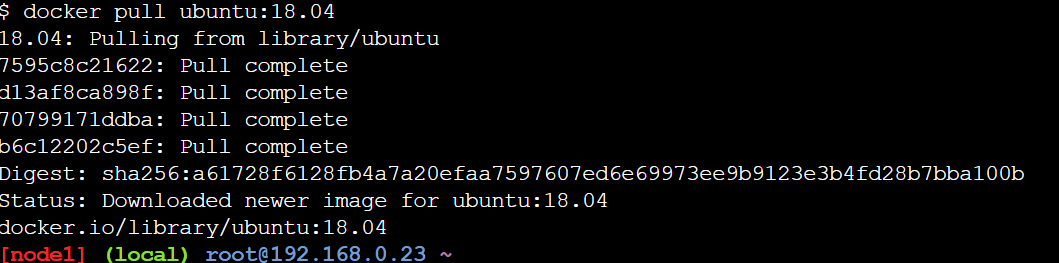
17. Find out which port is exposed for tomcat?

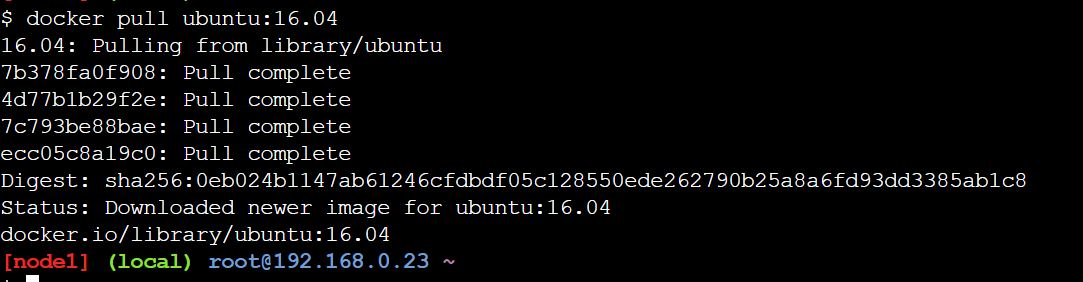
18. Run multiple conntainers of tomcat on different port and access it's home page.

docker run -d -p 8888:8080 tomcat

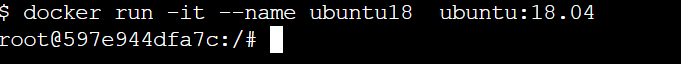
docker run -d -p 8888:8081 tomcat

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

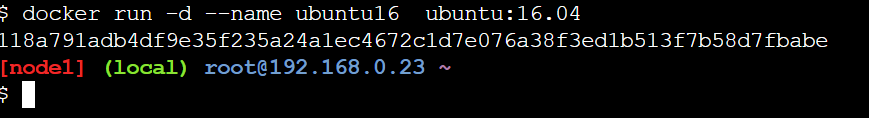




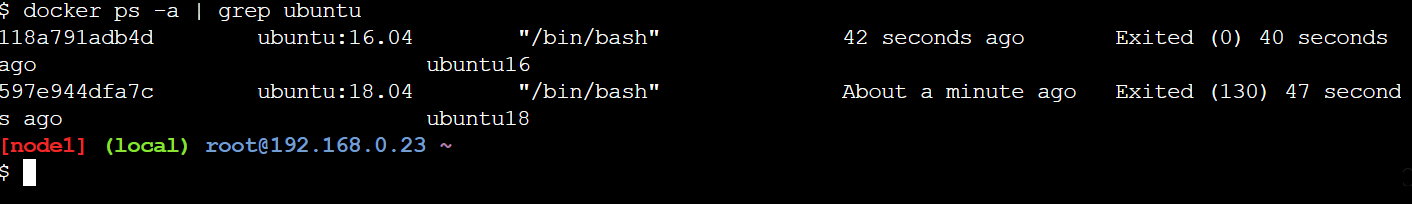
20. Run the container of ubuntu in attached mode.



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



22. Check how many ubuntu containers are running and stopped



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

docker start 7adef3512d6f

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 -f 7adef3512d6f

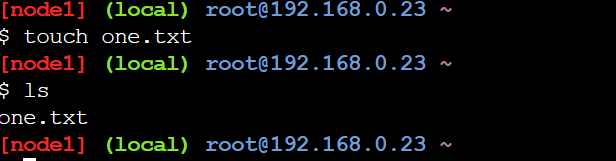
25. Check if ubuntu conatiner is running? If no, start one in attached mode to the terminal.

docker start 7adef3512d6f

26. Login as root user in ubuntu container

docker exec --user root -it 42d1c27fbb30 /bin/bash

27. Create a file with any name in root directory



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

apt-get update

apt-get install git

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 | grep ubuntu

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

docker run -it --name ubuntu18 -p 8081:8080 ubuntu:18.04

32. Login as a root user

docker exec --user root -it ubuntu18 /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.

docker rm <container id>

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

Not running

35. Create an Image out of the existing container.

docker commit 7f329ef97835 ubunutimage

36. Now Create a Container out of this image a

nd login into it to see if you can see the file and software installed by you in the previous container.

docker run -it --name newubuntu ubunutimage

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

docker stop <id>

docker kill <id>

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 -v /root/index.html:/usr/local/tomcat/webapps/ROOT/index.jsp --name tomcat tomcat:latest

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

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

docker images -a =>gives all images

docker images -q =>gives image id's

docker images -f=>gives filter options

for ex:-

docker images --filter "dangling=true" =>gives all the images which are not used by the containers

docker images --filter "before=tomcat:8"=>gives all the images which we used before tomcat:8

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

1. docker ps -a=>gives all containers which are stopped or exicted started

2. docker ps -f status=exicted=>gives all the continaers were exicted

3. docker ps -q=>gives all containerid's

4. docker ps -n 2=>gives last 2 contianers

5. docker ps -l=>gives latest continaer were running

6. docker ps -s=>gives the size of the continers used