

DOCKER 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 available options and commands to work with docker images

docker container --help ---> This command shows all the available 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.

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

Ans: a) docker search nginx

b)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 -a

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

Ans: REPOSITORY TAG IMAGE ID CREATED SIZE
nginx latest 87af4d3b9963 2 weeks ago 133MB

5. Create a container of your image

Ans: docker run -it --name PractContainer nginx

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: CONTAINER ID IMAGE COMMAND CREATED
STATUS PORTS NAMES
8fe6aff9716a nginx "/docker-entrypoin...." 7 minutes ago Up 6 minutes
80/tcp PractContainer

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

Ans: docker inspect nginx -6 layers

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

Ans: `docker pull tomcat:7`

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

Ans: `docker run -it --name MyTomcat --rm -p 8080:8080 tomcat:7`

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

Ans: `docker inspect MyTomcat -172.17.0.2`

13. Which Port it is using?

Ans: `8080/tcp`

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

Ans: `HTTP ERROR 404`

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

Ans: `533Mb`

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 inspect MyTomcat`

17. Find out which port is exposed for tomcat?

Ans : `docker inspect MyTomcat-8080`

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

Ans : `docker run -it --name MyTomcat --rm -p 8081:8080 tomcat:7.0`

`docker run -it --name MyTomcat --rm -p 8082:8080 tomcat:7.0`

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

Ans: `docker pull ubuntu:16.04`

`docker pull ubuntu:15.04`

20. Run the container of ubuntu in attached mode.

Ans: `docker run -it --name ubuntu16 f6f49faac5cf`

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

Ans: `docker run -it -d --name ubuntu15 d1b55fd07600`

22. Check how many ubuntu containers are running and stopped

Ans: `docker ps -a`

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

Ans: `docker run -it --name MyTomcat --rm -p 9090:8080 tomcat:7.0`

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 MyTomcat`

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

Ans: `docker ps=>docker run -it --name ubuntu16 f6f49faac5cf`

26. Login as root user in ubuntu container

Ans: `sudo -s`

27. Create a file with any name in root directory

Ans: `touch DockerAssgmnt`

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

Ans: `apt-get -y install sudo`

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

Ans: `exit` or `ctrl+d`

30. Check if the ubuntu container is running.

Ans: `docker ps`

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

Ans: `docker run -it --name newubuntu16 f6f49faac5cf`

32. Login as a root user

Ans: `sudo -s`

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: `ls`

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: `done`

35. Create an Image out of the existing container.

Ans: `docker commit 7dc7a14c6482 imgcontainer`

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 -it --name ContainerofImg 091829a52e94`

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.

39. type below command:-

```
docker images --help
```

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

1. -a = docker images -a == This will give all images including intermediate images
2. -f = docker images --filter "before=image1" == This will filter images created before the image 1(time based filter)
3. -q = docker images -q == lists only id's of the image not recommended unless a couple of images but can be used with --f tag to format along with tag or repo or time

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 == Shows both running and stopped containers
2. -f == docker ps --filter "name=containername" == filters based on provided container name
3. -q == docker ps -q -a == Shows only ids of all containers
4. -n == docker ps -n -l == shows last n containers along with their states
5. -l == docker ps -l == shows latest container
6. -s == docker ps -s == Shows total file sizes