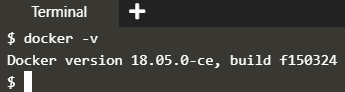
Assignment 1

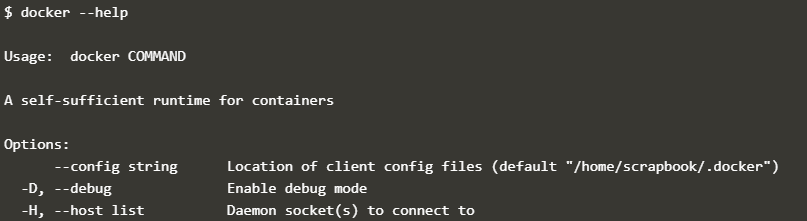
1. Install Docker, either on your native OS or on a VM. Make sure it runs. type "docker -v" to check if it's installed.

*docker –v*

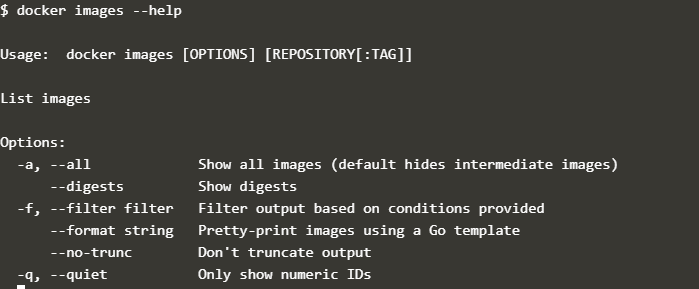


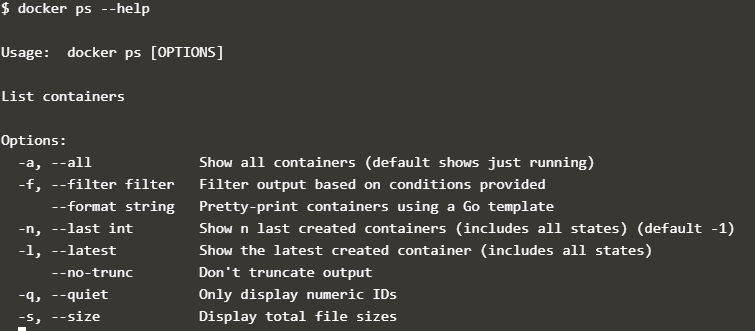
try below commands for help

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



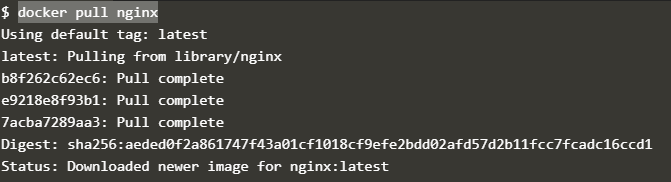
*docker images --help* ---> This command shows all the avaialble options and commands to work with docker images



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

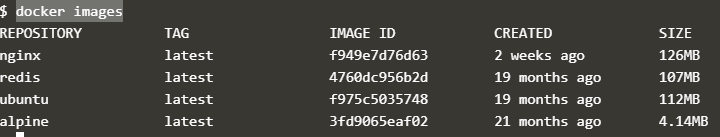
1. Find a image from dockerhub of your choice(recommeded: nginx), don't use browser, pull the official image from dockerhub

*docker pull nginx*



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

*docker images or docker image ls*



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

*docker images --no-trunc | grep nginx*

f949e7d76d63befffc8eec2cbf8a6f509780f96fb3bacbdc24068d594a77f043



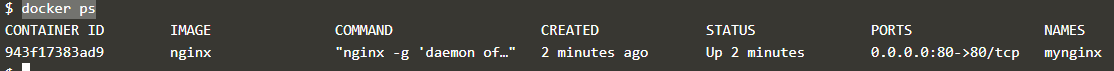
1. Create a container of your image

*docker run --name mynginx -p 80:80 -d nginx*



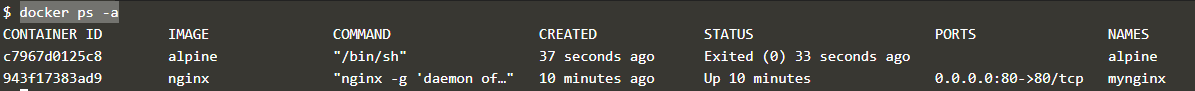
1. List all the running containers

*docker ps*



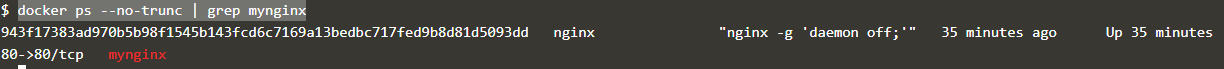
1. List all the running and stopped containers

*docker ps –a*



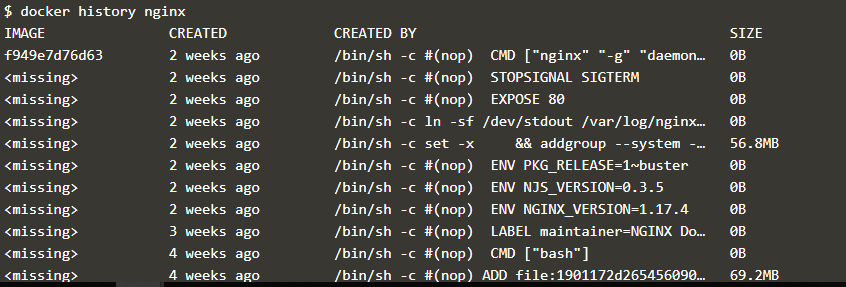
1. Find out the "Full" containerId of the container and write it below.

*docker ps --no-trunc | grep 943f17383ad9 or docker ps --no-trunc | grep mynginx*



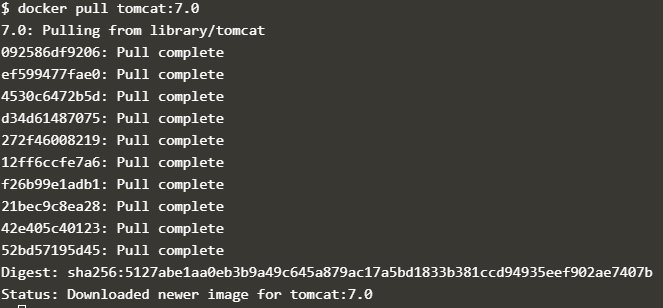
1. Find out how many image layers are used to build this image.

*docker history nginx*



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

*docker pull tomcat:7.0*



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

*docker run -d --name mytomcat tomcat:7.0*

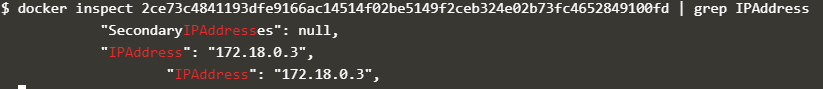


1. Find out what is the IP Address of the Apache Tomcat Container that it is running on

*docker inspect 2ce73c4841193dfe9166ac14514f02be5149f2ceb324e02b73fc4652849100fd | grep IPAddress*

or

*docker inspect 2ce73c4841193dfe9166ac14514f02be5149f2ceb324e02b73fc4652849100fd*



1. Which Port it is using?

*docker inspect 2ce73c484119*



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

Not getting correct ip address of katacoda playground.

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

*docker image ls | grep tomcat or docker images | grep tomcat*



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?

17. Find out which port is exposed for tomcat?

*docker inspect 2ce73c4841193dfe9166ac14514f02be5149f2ceb324e02b73fc4652849100fd | grep ExposedPort*

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

*docker run -d -p 8001:8080 --name mytomcat1 tomcat:7.0*

1373bd7ae911e46b2ecc62224c2ff19a6458cd510aef1bca3038a5ad72a6ee5e

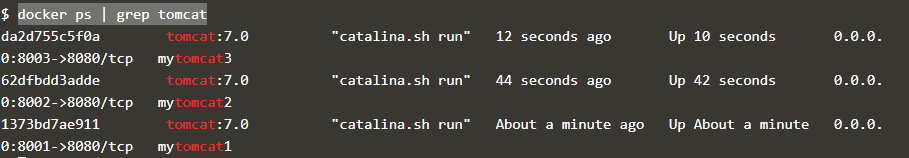
*docker run -d -p 8002:8080 --name mytomcat2 tomcat:7.0*

62dfbdd3adde762fa46cfd29c730a268f07e6063433c07e01da761b81ad04a4b

*docker run -d -p 8002:8080 --name mytomcat2 tomcat:7.0*

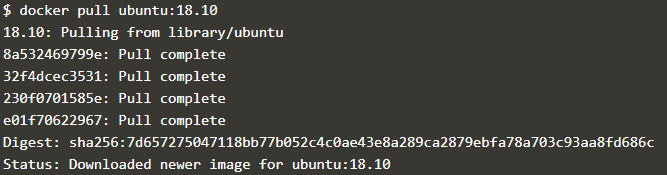
da2d755c5f0abe5fd6d105f0f7c48a00dba63709c60d34b8f82b19b02bc09e96

*docker ps | grep tomcat*

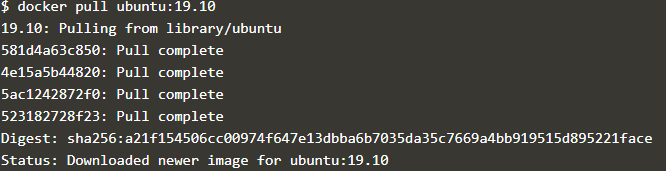


19. Pull ubuntu os from dockerhub, try to pull 2 images of ubuntu, Except the latest one.

*docker pull ubuntu:18.10*

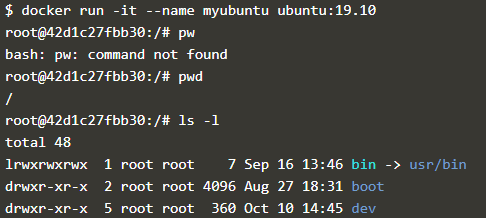


*docker pull ubuntu:19.10*



20. Run the container of ubuntu in attached mode.

*docker run -it --name myubuntu ubuntu:19.10*



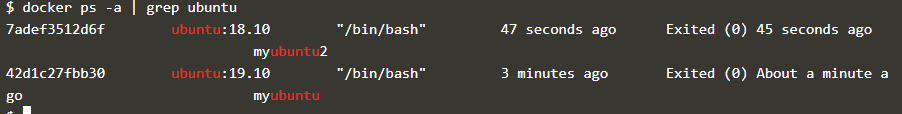
21. Run the container of another ubuntu in detached mode.

*docker run -d --name myubuntu2 ubuntu:18.10*



22. Check how many ubuntu containers are running and stopped

*docker ps -a | grep ubuntu*



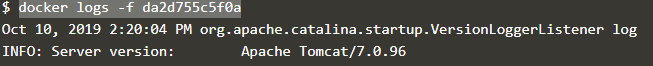
23. Is the tomcat container running? If no, start one.

*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 da2d755c5f0a*



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

*docker start --attach 42d1c27fbb30*



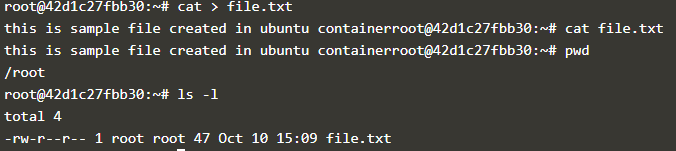
26. Login as root user in ubuntu container

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



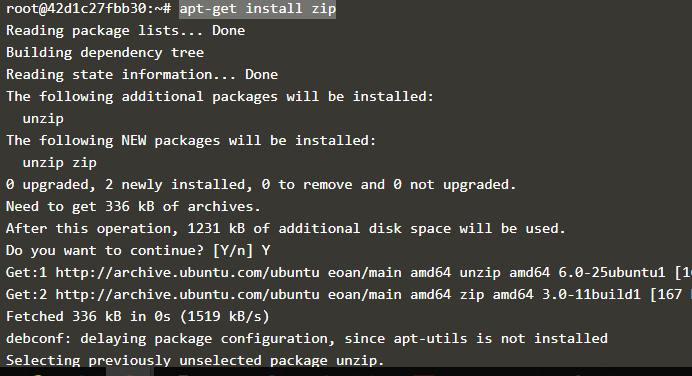
27. Create a file with any name in root directory

*cat > file.txt*



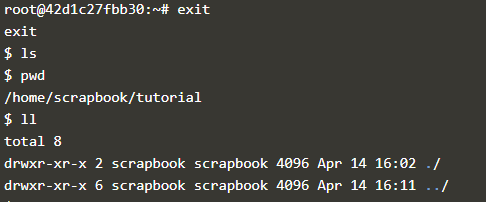
28. Install software of your choice in ubuntu container using "apt-get install"

*apt-get install zip*



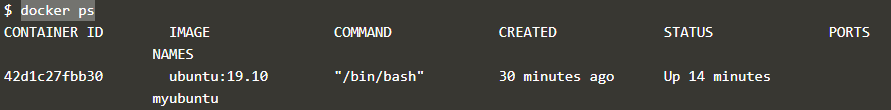
29. Now exit the ubuntu shell, are you back to your host machine, if not, come back to the host machine.

*root@42d1c27fbb30:~# 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 myubuntu2 ubuntu:19.10*



32. Login as a root user

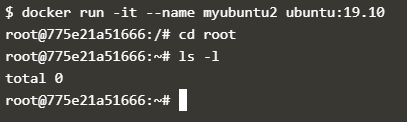
In attached mode by default it is logging with root

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.

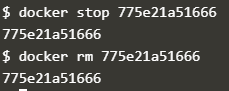
File not available as this is new container

*cd root*

*ls –l*



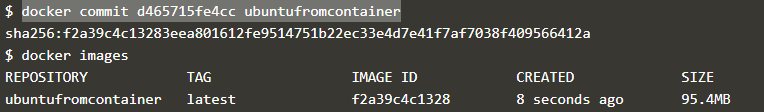
To stop container and remove the container



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

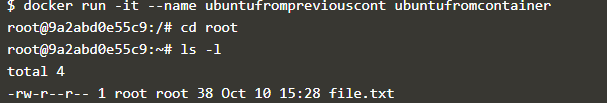
35. Create an Image out of the existing container.

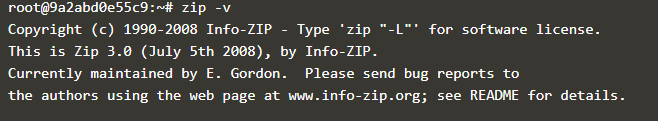
*docker commit d465715fe4cc ubuntufromcontainer*



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 ubuntufrompreviouscont ubuntufromcontainer*





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

*docker stop 9bdf17cfa3fa*

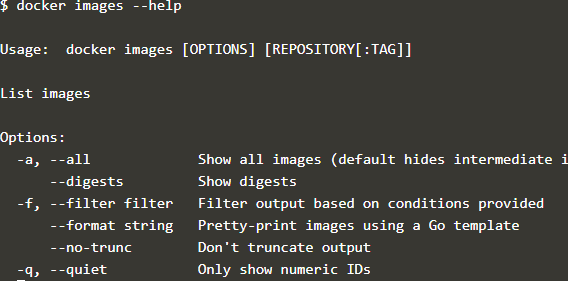
38. Create an index.html file with following code in it:-

<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.

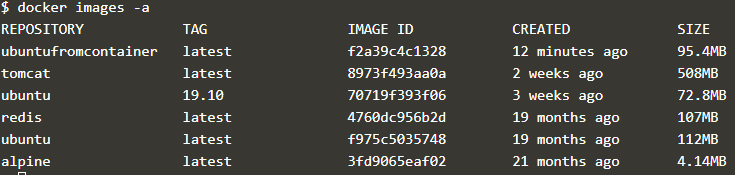
39. type below command:-

*docker images –help*



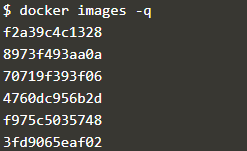
Now, try to run command that proves the concept of following three options:-

1. –a



1. –f not done

3. -q



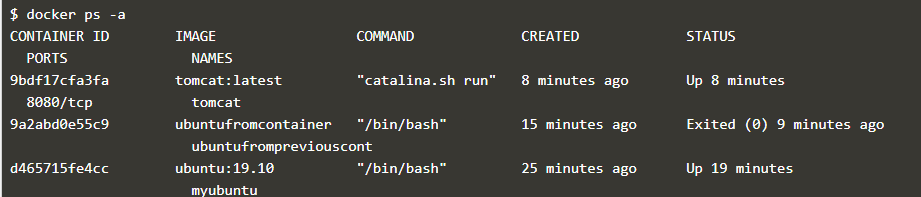
write at least 1 command using each option above and prove their concepts as described in the --help.

40. type below command:-

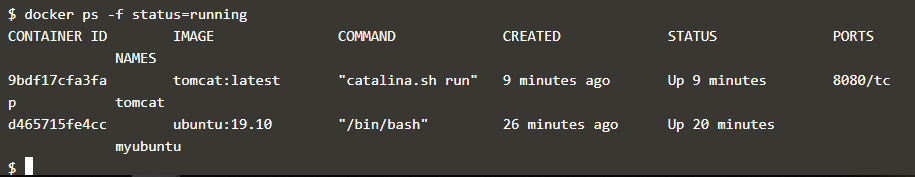
*docker ps --help*

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

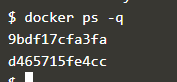
1. –a



1. –f



1. –q only running container



4. -n

5. –l



6. -s

