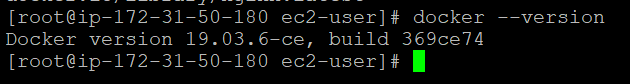
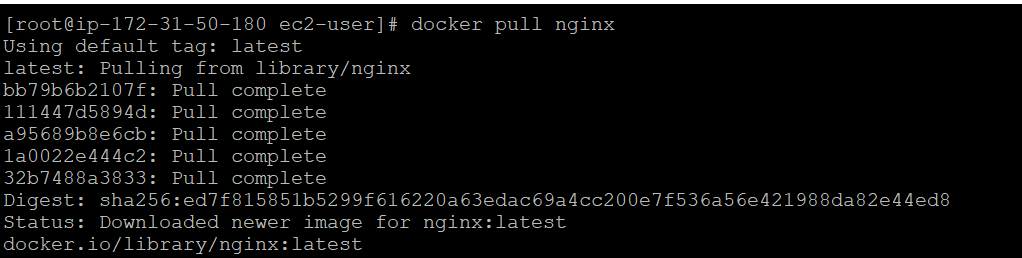
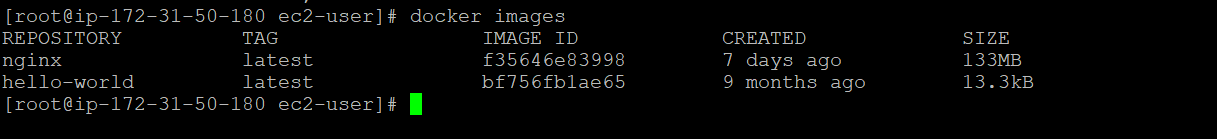
1. Install Docker, either on your native OS or on a VM. Make sure it runs. Type "docker -v" to check if it's installed.



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

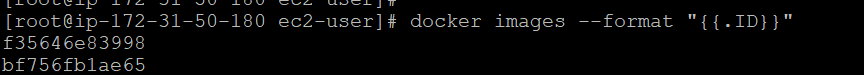


1. List all the available images in your machine/vm, make sure you see recently pulled image in the list.



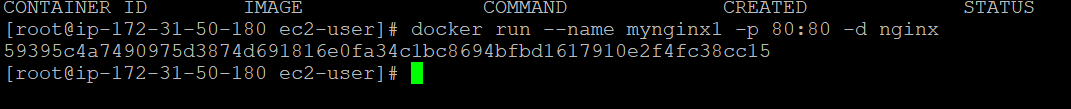
1. Find out the "Full" ImageId of the image that you pulled and write it below.

docker images --format "{{.ID}}



1. Create a container of your image

docker run --name mynginx1 -p 80:80 -d nginx

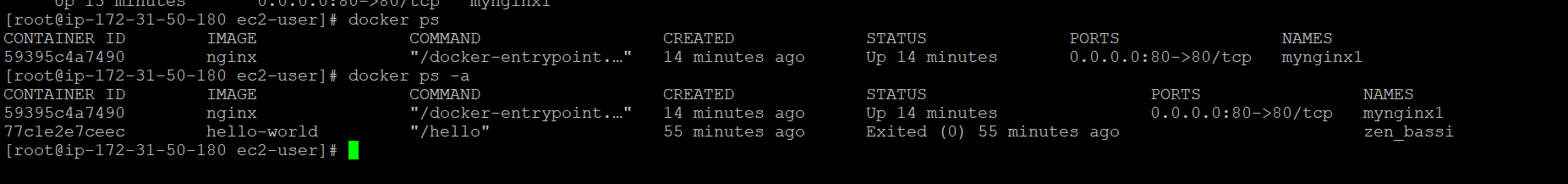


1. List all the running containers

docker ps

1. List all the running and stopped containers

Docker ps –a



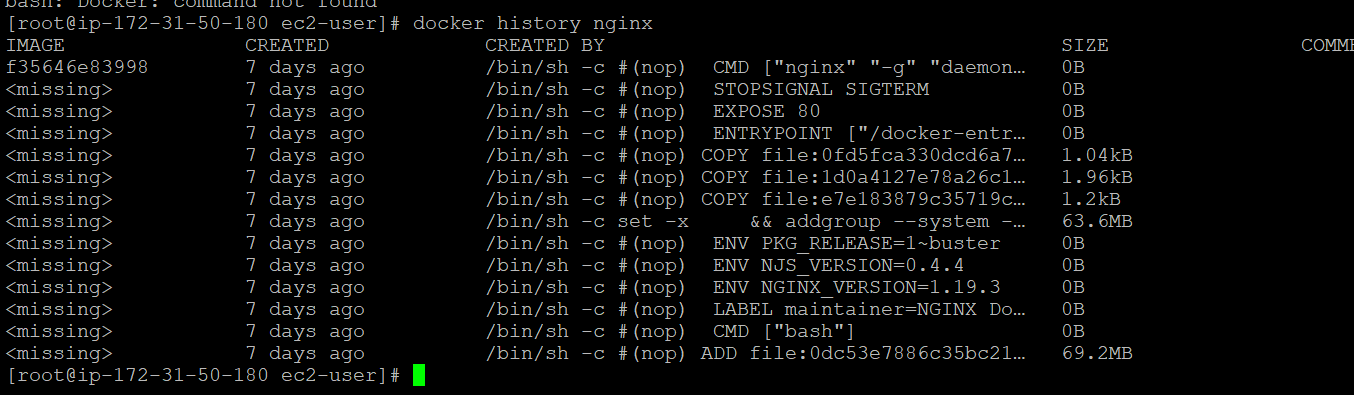
1. Find out the "Full" containerId of the container and write it below.

docker inspect --format="{{.Id}}" container\_name



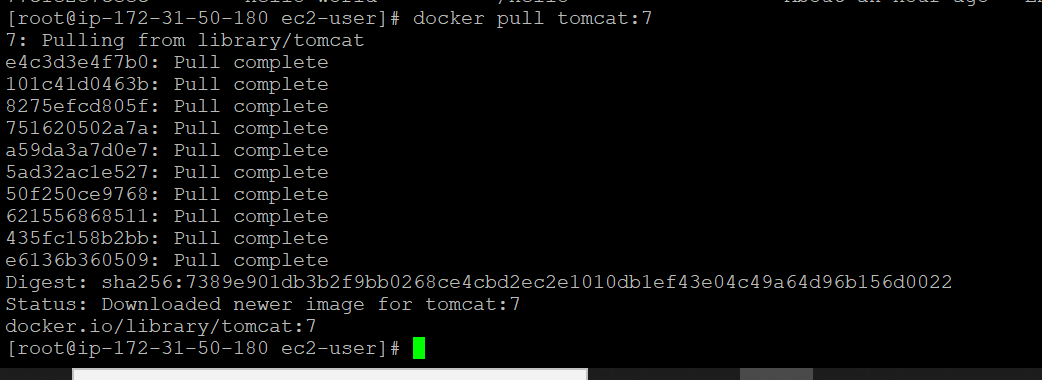
9.Find out how many image layers are used to build this image.

Docker history nginx



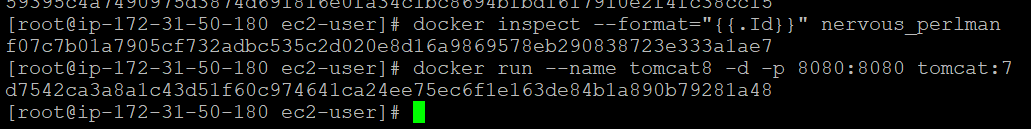
1. Get the Apache Tomcat 7 server image from the docker hub.

docker pull tomcat:7



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

docker run --name tomcat8 -d -p 8080:8080 tomcat:7



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

Docker inspect container id

1. Which Port it is using?

Docker inspect container id

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

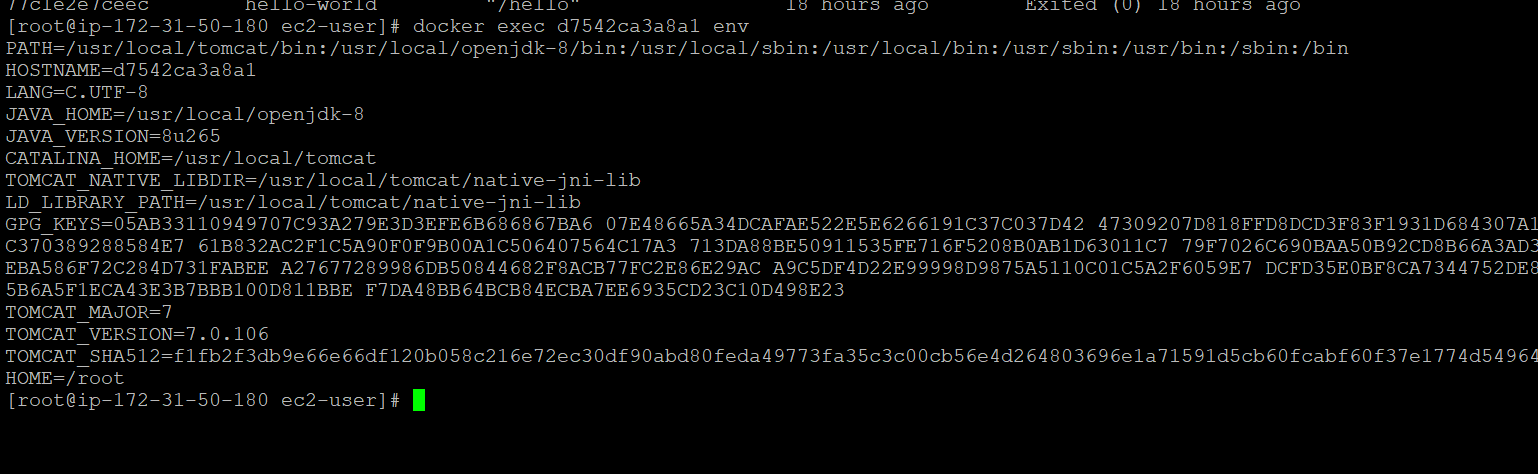
<http://ipaddress:8080>

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

Docker ps –size

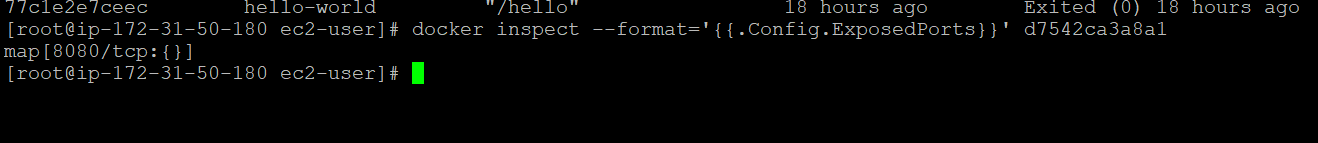
1. Find out list of all environment variables that is configured for tomcat image, can you see JAVA\_HOME and CATALINA\_HOME? What did you notice about it?

Docker exec container id env



1. Find out which port is exposed for tomcat?

docker inspect --format='{{.Config.ExposedPorts}}' container id



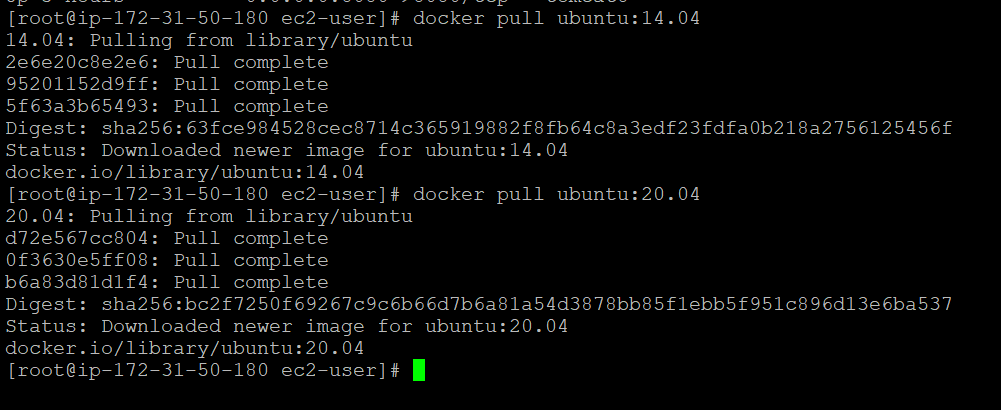
1. Run multiple conntainers of tomcat on different port and access it's home page.

docker run -d -p 8888:8080 tomcat

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

docker pull ubuntu:14.04

docker pull ubuntu:20.04

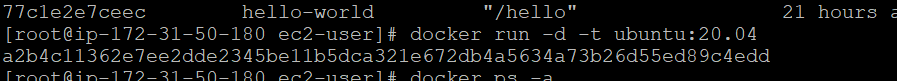


1. Run the container of ubuntu in attached mode.

Docker attach Ubuntu:20.04

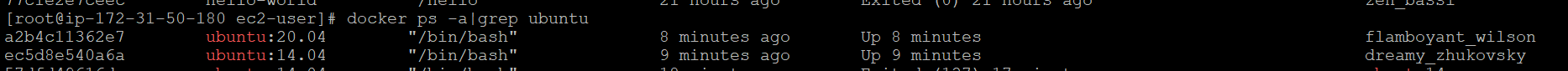
1. Run the container of another ubuntu in detached mode.

docker run -d -t ubuntu:14.04



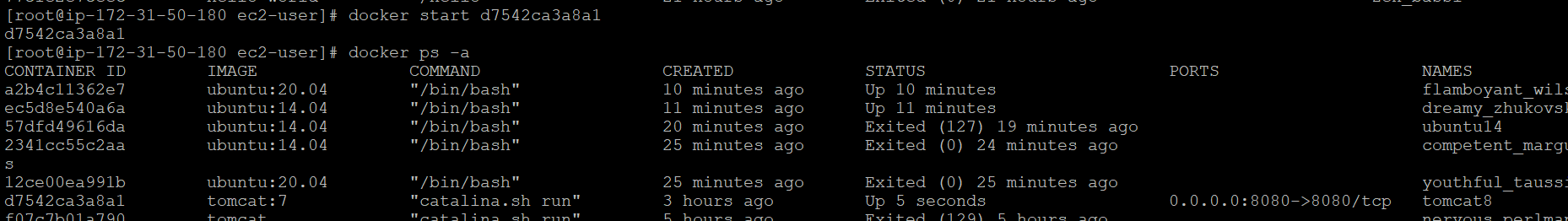
1. Check how many ubuntu containers are running and stopped

docker ps –a|grep Ubuntu



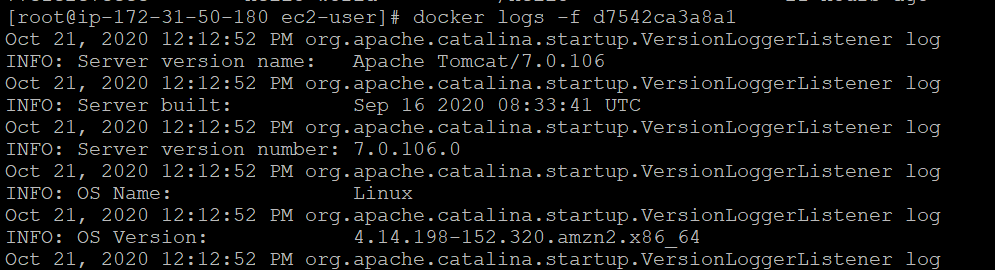
1. Is the tomcat container running? If no, start one.

docker start container id



1. 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 container id

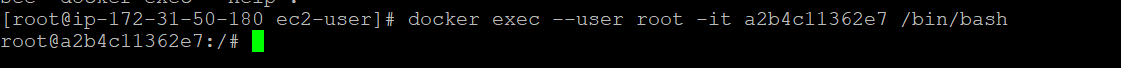


1. Check if ubuntu conatiner is running? If no, start one in attached mode to the terminal

docker start container id

1. Login as root user in ubuntu container.

docker exec --user root -it a2b4c11362e7 /bin/bash



1. Create a file with any name in root directory

Touch root.txt

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

Apt-get update

Apt-get install vim

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

exit

1. Check if the ubuntu container is running.

docker ps|grep Ubuntu

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

docker run -it --name ubuntu14 -p 8081:8080 ubuntu:14.04

1. Login as a root user

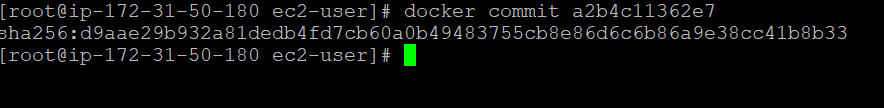
docker exec --user root -it flamboyant\_wilson /bin/bash

1. 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 rmi container id

1. Do you have the previous ubuntu container where you created the file and installed the software? If no reapeat step 25 to 29.
2. Create an Image out of the existing container.

Docker commit container id new name



40. 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