Running containers

• Enter the docker container run command in a Bash terminal. This instructs Docker to run a container called hello-world:

1. $ docker container run hello-world

A: Docker looked for hello-world file locally(with in the system), therefor it could not find it, it went to public repo docker hub to see if there's any related to hello world, once it found the file, it pulled(downloaded) from docker repo, displayed the hello message what it has in hello-world image, gave additional links to refer and exited from docker hub.

Use the docker container ls command to see what containers are running on your system. In your Bash terminal, type the following command:

2. $ docker container ls

A: Command shows no of containers in the docker container, as of now there are none in my container.
Shows columns like : CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

Use the docker container ls -a command to display all the containers, even the stopped ones:

3. $ docker container ls -a

What output do you get this time on your screen?

A: Shows stopped hello-world container that ran and stopped in lab3 in 3rd week, with the detailed information of container ID, image name, command, created time, status, ports, names.

You can query your system to see what container images Docker cached locally. Execute the docker image ls command to view the local cache:

4. $ docker image ls

What is the image tag associated with your hello-world container?

A: The tag is the latest and image tag associated is **"feb5d9fea6a5"**

CONTINUE TO NEXT PAGE

Managing container life cycles

• In your terminal, execute the docker image pull command to download the Ubuntu 18.04 container image:

$ docker image pull ubuntu:18.04
What does the 18.04 represent in the preceding command?

A: 18.04 defines the version number for ubuntu release. Say ubuntu18.04 is the operating system and 18.04 is the release version. It can be LTS or with temporary support.

•Use the docker image ls command to confirm that the container images are downloaded to the local container cache:
$ docker image ls
What are the contents of the local container cache?

A: Contents of local container cache are ubuntu 18.04, 19.04 images and hello-world containers, with image id's e28a50f651f9, c88ac1f841b7 and feb5d9fea6a5

•$ docker image inspect IMAGE ID – 18.04
When was this image created?

A: Image Created: "2023-01-02T18:48:56.081327405Z"

•$ docker image inspect IMAGE ID – 19.
When was this image created?

A: Image Created: "2020-01-16T01:20:46.938732934Z"

What can you conclude after inspecting both container images?

A: Both have different images ID's, sizes, have different **docker versions**, different repo digests, mainly they both has different RootFS layers where ubuntu18.04 has only 1 sha layer, while 19.04 has 4 sha layers. Different id's, **hostname, image** that gives us information that both ubuntu versions are from different systems/repo's, containers. While ubuntu 18.04 has cmd as "bash", 19.04 has "/bin/bash" which makes me wonder, that they both different project root folders. Minor difference was in 19.04 has "container config" has key value pair "ArgsEscaped":true while 18.04 doesn't, that tells us 19.04 is skipping container config ArgsEscaped which may be a upgrade from 18.04 to 19.04.

Check the status of the container using the docker container ls -a command:
•$ docker container ls -a
What is the status of your ubuntu container and explain?

A: Status of ubuntu container is "exited (0) 2 seconds ago". When we run the command "docker container run -d ubuntu:18.04 ", it ran immediately the ubuntu image, went in, got the required information and id, image name, creation of container, exited by gathering command and hostname of the repo

•$ docker container exec -it ubuntu1 /bin/bash
What is the output of your terminal? What is the hostname of your container and where is it taken from?
A: root@a2f62cfab86e:/#, hostname is a2f62cfab86e, it is taken from container ID of ubuntu1

•$ docker container exec -it ubuntu1 cat hello-world.txt
Run the same cat command in the ubuntu2 container instance:
•$ docker container exec -it ubuntu2 cat hello-world.txt
Explain in detail what just happened? Why were you moved back to the context of your main terminal in both cases?

A: From last 4 commands we are went to container using "docker container exec" command to access ubuntu OS 18.04,19.04 version and created a file called hello-world.txt with a command in bash terminal of ubuntu "echo" to print a message. After exiting from both containers, we tried access the container using exec. Cat is a command that prints the what ever the text file has, so that we can print what ever hello-world text files has.

•$ docker container stop ubuntu2
Use the docker container ls command to view all running container instances:
$ docker container ls
Which container is up and running?

A: container with image ubuntu:18.04 is up and running which is ubunt1

•$ docker container ls -a
What is the output of the preceding command?
Answer: Shows all containers created, used, up and running and stopped. Below is the output

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
98bfa47dc4b8 ubuntu:18.04 "bash" 21 minutes ago Exited (0) 2 seconds ago ubuntu4
7eafffda54fa ubuntu:19.04 "/bin/bash" 26 minutes ago Exited (0) 5 minutes ago ubuntu3
d4a03ecfde3f ubuntu:18.04 "bash" About an hour ago Exited (0) About an hour ago ubuntu2
a2f62cfab86e ubuntu:18.04 "bash" About an hour ago Up 15 seconds ubuntu1
8755cc15d078 ubuntu:18.04 "bash" 3 hours ago Exited (0) 3 hours ago unruffled_galileo
a649f5b832dd ubuntu:18.04 "bash" 3 hours ago Exited (0) 3 hours ago bold_hellman
2257b2f66b46 hello-world "/hello" 31 hours ago Exited (0) 31 hours ago gallant_margulis

•Verify that the container is running again by using the docker container ls command:
•$ docker container ls
What can you conclude about the "Status" of the newly started/restarted container?

A: Status says the timeline of container since it started, be it seconds, minutes, hours.

Execute docker container ls -a to see all containers.
•$ docker container ls -a
What is the output of your display?

A: Shows what are remaining containers in docker. Below is the output

```
8755cc15d078   ubuntu:18.04   "bash"     3 hours ago    Exited (0) 3 hours ago          unruffled_galileo
a649f5b832dd   ubuntu:18.04   "bash"     3 hours ago    Exited (0) 3 hours ago          bold_hellman
2257b2f66b46   hello-world    "/hello"   32 hours ago   Exited (0) 32 hours ago         gallant_margulis
```

What happened to all the other containers you worked with?
A: They are deleted from docker when removed containers using rm command from our host(ec2 instance) which were installed previously.