Containers - Docker - Exercises

I - Installations

Refer to the following link to download Docker Desktop for your specific platform:
https://docs.docker.com/engine/install/

Run the installation program.

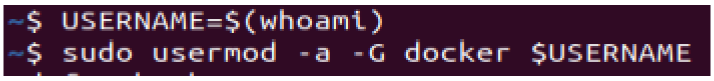
Once docker is installed, you should be able to run the Docker Daemon.

Configuring Permissions on Unix systems:

If you're using Unix, you may face a "Permission Denied" problem :
Docker: Got permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock

This means your user doesn't have privileges to interact with the docker daemon (dockerd)

If this happens, type the following commands in your terminal:



```
~$ USERNAME=$(whoami)
~$ sudo usermod -a -G docker $USERNAME
```

Restart your terminal then type the **groups** command and make sure **docker** appears in the list.

Now if you type docker commands, the docker daemon should respond without error.

II - Practicing

A - Hello world

In this first exercise, we'll use some basic docker commands to get started.

- First of all, let's list the images available in our local registry:

$ docker images

- Then we'll pull the ubuntu 20.04 image :

$ docker pull ubuntu:20.04

- List the available images list again.

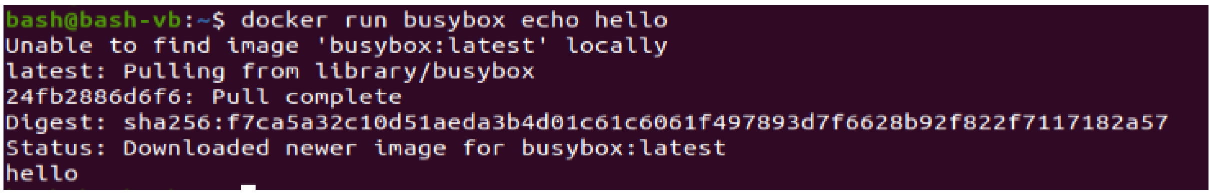
How can you explain the image size is only about 70MB while the distribution is 3GB ?

- Now, let's run a lightweight image :

$ docker run busybox echo hello

Containers - Docker - Exercises

Note that docker pulled the image since it did not exist in our local registry before executing our command !



```
bash@bash-vb:~$ docker run busybox echo hello
Unable to find image 'busybox:latest' locally
latest: Pulling from library/busybox
24fb2886d6f6: Pull complete
Digest: sha256:f7ca5a32c10d51aeda3b4d01c61c6061f497893d7f6628b92f822f7117182a57
Status: Downloaded newer image for busybox:latest
hello
```

- Try to remove the busybox image. How can we do that ?

Useful commands :
docker ps lists the running containers
docker ps -a lists all the containers (running / stopped / killed / …)
docker rm <id> removes the specified inactive container
docker rmi <id> removes the specified unused image

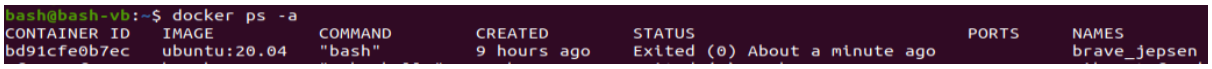
- Run an Ubuntu container interactively:

$ docker run -it ubuntu:20.04

In your new container, install python3:
/# apt-get update && apt-get install -y python3

If the update fails, check that your VM' system date is correct. Otherwise fix it.

- Run exit on your container's terminal.
- Showing running processes should return an empty list while showing all processes should list the container you just stopped.



```
bash@bash-vb:~$ docker ps -a
CONTAINER ID   IMAGE          COMMAND        CREATED        STATUS                          PORTS     NAMES
bd91cfe0b7ec   ubuntu:20.04   "bash"         9 hours ago    Exited (0) About a minute ago             brave_jepsen
```

- We will restart it in order to install pip too

$ docker start bd91 #bd91 should be replaced with your container ID's first characters.
$ docker exec -it bd91 bash # here we are executing interactively the bash command in the bd91 container.

/# apt-get install -y pip
/# exit

- What if you run docker ps ?
- Commit the container in order to create a new image

$ docker commit bd91 ubuntu-python3:latest
$ docker images

- Stop and destroy the bd91 container
- Create a new container from your new image and make sure python3 and pip commands are installed

$ docker run -it ubuntu-python3

Containers - Docker - Exercises

B - Docker tutorial

- Run the following command:

```
$ docker run -d -p 80:80 docker/getting-started
```

This will run a webapp, accessible from the url http://localhost in your VM.

- Follow the tutorial. Note that Docker Dashboard is only available on Mac / Windows. We will use the Command Line Interface.

C - More Docker with Katas series

Clone the following repository :
https://github.com/eficode-academy/docker-katas/tree/master/labs
The exercises are described in the markdown files.

D - Building Images & composing services

1. Building an SQL Lab environment

- Create Dockerfiles for multiple services:
    - A postgresql Database with pre-initialized data (ddl + dml).
    - A Jupyter Notebook with some modules :
        - to run SQL queries.
        - to interact with Github (ie: https://github.com/jupyterlab/jupyterlab-git)
- Create a Compose file which builds and runs both services.
- Start the services.
- Create a notebook and check everything's working as expected.
- Using the table `SQL_AVANCE.T_FILMS`, how many movies are there by gender ? (column **ls_genre**)
- Add a volume to your compose file in which you can store your notebooks.

2. Building a search service environment

- Download the following archive :

https://storage.cloud.google.com/baitmbarek-dataops/movies-db-bulk.zip

- Clone the following repository :

https://github.com/baitmbarek/dataops-pyhton-movie-search.git

In this exercise, you'll have to setup an environment with at least two Dockerfiles:

- An elasticsearch server with an index containing the movies from the downloaded dataset. You can create the index with a command similar to the following:

```
curl -H 'Content-Type: application/json' -XPOST http://elastic:9200/_bulk --data-binary
```

@/path/to/movies-db-bulk.json

- An application running the cloned code. Note that the application uses both Flask and Elasticsearch. By default, it listens on port 8000.

A docker-compose up command should run the whole environment.

E - Quizz

- **1.** Docker containers are based on open standard ______________.
    - A. Allowing containers to run on all major Linux distributions only
    - B. Allowing containers to run on all Microsoft operating systems only
    - C. Allowing containers to run on all major Linux distributions and Microsoft operating systems
- **2.** Containers running on a single machine all share the same operating system kernel, so they start instantly and make more efficient use of RAM.
    - A. True
    - B. False
- **3.** Virtual Machines, Each virtual machine includes the application, the necessary binaries and libraries, and an entire guest operating system - All of which may be tens of GBs in size.
    - A. True
    - B. False
- **4.** Containers; Containers include the application and all of its dependencies, but share the kernel with other containers. They run as an isolated process in userspace on the host operating system. They're also not tied to any specific infrastructure – Docker containers run on any computer, on any infrastructure, and in any cloud.
    - A. True
    - B. False
- **5.** _______is a cloud-hosted service from Docker that provides registry capabilities for public and private content.
    - A. Docker Swarm
    - B. Docker Hub
    - C. Docker Cloud
    - D. Docker Compose
- **6.** _______is a tool for defining and running multi-container Docker applications.
    - A. Docker Swarn

- B. Docker Hub
- C. Docker Cloud
- D. Docker Compose

- **7.**________is a text document that contains all the commands a user could call on the command line to assemble an image.
  - A. Dockerfile
  - B. Docker Compose
  - C. Makefile
- **8.** On Docker Hub, you get ten private repositories for free with your Docker Hub user account. If you need more accounts, you can upgrade your Docker Hub plan.
  - A. True
  - B. False
- **9.** Docker host's IP address by default is 172.17.0.254
  - A. True
  - B. False
- **10.** Following Docker command: docker exec -it **container_id** bash is used to:
  - A. Activate default VM machine
  - B. Access a running container
  - C. Build an image
  - D. Commit changes done in a Docker image
- **11.** Following Docker command: docker build -t my_user/repo_name:1.0 is used to:
  - A. Activate default VM machine
  - B. Access a running container
  - C. Build an image
  - D. Commit changes done in a Docker image
- **12.** Following Docker command: docker commit -m "My first update" **container_ID user_name**/**repository_name** is used to:
  - A. Activate default VM machine
  - B. Access a running container
  - C. Build an image
  - D. Commit changes done in a Docker image
- **13.** Following Docker command: docker push user_name/repository_name is used to:
  - A. Activate default VM machine
  - B. Push changes done in an docker image into Docker Hub
  - C. Build an image

- D. Commit changes done in a Docker image