NAME: AAKASH NAMALA **ROLL NO: 20A91A0544**

GITHUB LINK: https://github.com/aakash-namala/Hero_Vired.git

Question 1:

Pull any image from the docker hub, create its container, and execute it showing the output.

To pull the image from the Docker Hub we need to use the docker pull command.

Syntax: docker pull image_name

Then the docker daemon will interact with the docker hub and start downloading the images.

```
C:\Users\Aakash007>docker login
Login with your Docker ID to push and pull images from Docker Hub. If you don't have a Docker ID, head over to
 https://hub.docker.com to create one.
Username: aakash7123
Password:
Login Succeeded

Logging in with your password grants your terminal complete access to your account.
For better security, log in with a limited-privilege personal access token. Learn more at https://docs.docker.
com/go/access-tokens/

C:\Users\Aakash007>
```

```
C:\Users\Aakash007>docker pull mysql
Using default tag: latest
latest: Pulling from library/mysql
197c1adcd755: Downloading  41.59MB/44.56MB
45f2e353f7d2: Download complete
68ec6ece42ef: Download complete
cfa4d9a7b88e: Download complete
64cab5858b1d: Download complete
92fcd248d982: Download complete
88635e83312d: Downloading  22.58MB/56.22MB
43f0427259d9: Download complete
79828698a290: Downloading  11.39MB/46.34MB
a8854781893e: Waiting
6c8bdf3091d9: Waiting
```

```
C:\Users\Aakash007>docker pull mysql
Using default tag: latest
latest: Pulling from library/mysql
197c1adcd755: Pull complete
45f2e353f7d2: Pull complete
68ec6ece42ef: Pull complete
cfa4d9a7b88e: Pull complete
64cab5858b1d: Pull complete
92fcd248d982: Pull complete
88635e83312d: Pull complete
43f0427259d9: Pull complete
79828698a290: Pull complete
a8854781893e: Pull complete
6c8bdf3091d9: Pull complete
Digest: sha256:8653a170e0b0df19ea95055267def2615fc53c62df529e3750817c1a886485f0
Status: Downloaded newer image for mysql:latest
docker.io/library/mysql:latest
```

The docker pull only download the images, it won't create containers or start the images.

```
C:\Users\Aakash007>docker pull mysql
Using default tag: latest
latest: Pulling from library/mysql
197c1adcd755: Pull complete
45f2e353f7d2: Pull complete
68ec6ece42ef: Pull complete
cfa4d9a7b88e: Pull complete
64cab5858b1d: Pull complete
92fcd248d982: Pull complete
88635e83312d: Pull complete
43f0427259d9: Pull complete
79828698a290: Pull complete
a8854781893e: Pull complete
6c8bdf3091d9: Pull complete
Digest: sha256:8653a170e0b0df19ea95055267def2615fc53c62df529e3750817c1a886485f0
Status: Downloaded newer image for mysql:latest
docker.io/library/mysql:latest

C:\Users\Aakash007>docker images
REPOSITORY               TAG       IMAGE ID       CREATED         SIZE
java_app                 latest    b4bace44ec0d   2 days ago      526MB
nginx                    latest    3f8a00f137a0   10 days ago     142MB
mysql                    latest    57da161f45ac   10 days ago     517MB
ubuntu                   latest    58db3edaf2be   3 weeks ago     77.8MB
docker/getting-started   latest    3e4394f6b72f   8 weeks ago     47MB
resin/docs               latest    592de848a9b7   4 months ago    1.1GB
hello-world              latest    feb5d9fea6a5   17 months ago   13.3kB

C:\Users\Aakash007>
```

Nginx image is present in images list. But it is not present in the containers list.

```
C:\Users\Aakash007>docker ps --all
CONTAINER ID   IMAGE                    COMMAND                  CREATED        STATUS                      PORTS               NAMES
1c2f9155fcff   ubuntu                   "/bin/bash"              39 hours ago   Exited (255) 36 hours ago                       funny_bassi
77e8440341ae   hello-world              "/hello"                 4 days ago     Exited (0) 4 days ago                           nervous_franklin
10ec49f9859c   docker/getting-started   "/docker-entrypoint.…"   5 days ago     Exited (255) 4 days ago     0.0.0.0:80->80/tcp   wizardly_turing

C:\Users\Aakash007>
```

To run the images, we need to create the containers. For this we can use the docker create command.

The docker create command is used to create a new container from an image without starting it. This command is useful when you want to set up a container in advance and then start it later using the docker start command.

Syntax: docker create [OPTIONS] IMAGE [COMMAND] [ARG...]

--name: This option allows you to specify a name for the container.

-e: This option allows you to set environment variables for the container.

-v: This option allows you to create a volume to share data between the host and the container.

```
C:\Users\Aakash007>docker create --name aakash_sql mysql
76febf19786ef3c757b837ef0b15a7290753f0bb2ecae5ad25f92fd524799945

C:\Users\Aakash007>docker ps --all
CONTAINER ID   IMAGE                    COMMAND                  CREATED          STATUS                      PORTS               NAMES
76febf19786e   mysql                    "docker-entrypoint.s…"   31 seconds ago   Created                                         aakash_sql
1c2f9155fcff   ubuntu                   "/bin/bash"              39 hours ago     Exited (255) 36 hours ago                       funny_bassi
77e8440341ae   hello-world              "/hello"                 4 days ago       Exited (0) 4 days ago                           nervous_franklin
10ec49f9859c   docker/getting-started   "/docker-entrypoint.…"   5 days ago       Exited (255) 4 days ago     0.0.0.0:80->80/tcp   wizardly_turing

C:\Users\Aakash007>
```

Now to start the container we need to use the start command.

Syntax: docker start container_id

```
C:\Users\Aakash007>docker start 76febf19786e
76febf19786e
```

```
C:\Users\Aakash007>docker run mysql
2023-02-19 12:43:04+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.0.32-1.el8 started.
2023-02-19 12:43:05+00:00 [Note] [Entrypoint]: Switching to dedicated user 'mysql'
2023-02-19 12:43:05+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.0.32-1.el8 started.
2023-02-19 12:43:05+00:00 [ERROR] [Entrypoint]: Database is uninitialized and password option is not specified
    You need to specify one of the following as an environment variable:
    - MYSQL_ROOT_PASSWORD
    - MYSQL_ALLOW_EMPTY_PASSWORD
    - MYSQL_RANDOM_ROOT_PASSWORD

C:\Users\Aakash007>
```

Or simply we can use the docker run command.

The docker daemon will search for the image in the local machine, if it does not find it, then it(it acts as an hypervisor b/w client and docker hub) will reach the docker hub and pull the image from the docker hub to the local machine, then it will create the container automatically and the then run the image.

Docker run => docker pull + docker create + docker start.

```
C:\Users\Aakash007>docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
2db29710123e: Pull complete
Digest: sha256:6e8b6f026e0b9c419ea0fd02d3905dd0952ad1feea67543f525c73a0a790fefb
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/
```

```
 https://docs.docker.com/get-started/


C:\Users\Aakash007>docker ps --all
CONTAINER ID   IMAGE                  COMMAND                  CREATED              STATUS                          PORTS                  NAMES
5b82c2756b91   hello-world            "/hello"                 About a minute ago   Exited (0) About a minute ago                          musing_meninsky
ddf1e0389281   mysql                  "docker-entrypoint.s…"   3 minutes ago        Exited (1) 3 minutes ago                               priceless_solomon
a2c6b2bea56c   mysql                  "docker-entrypoint.s…"   18 minutes ago       Exited (1) 18 minutes ago                              gracious_darwin
1c2f9155fcff   ubuntu                 "/bin/bash"              39 hours ago         Exited (255) 36 hours ago                              funny_bassi
10ec49f9859c   docker/getting-started "/docker-entrypoint.…"   5 days ago           Exited (255) 4 days ago         0.0.0.0:80->80/tcp     wizardly_turing

C:\Users\Aakash007>
```

Question 2:

Create the basic java application, generate its image with necessary files, and execute it with docker.

Create a directory for your java application.

Syntax: mkdir app_name

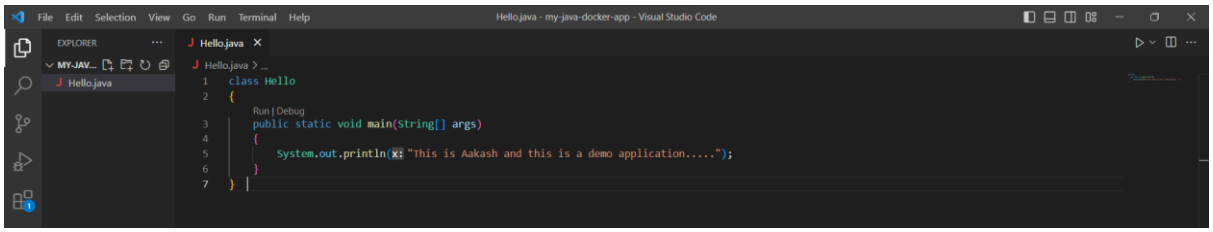
```
C:\Users\Aakash007>mkdir my-java-docker-app

C:\Users\Aakash007>
```

Now move to the created directory using the cd command. Then type “code .”, it will open your default editor for coding purpose.

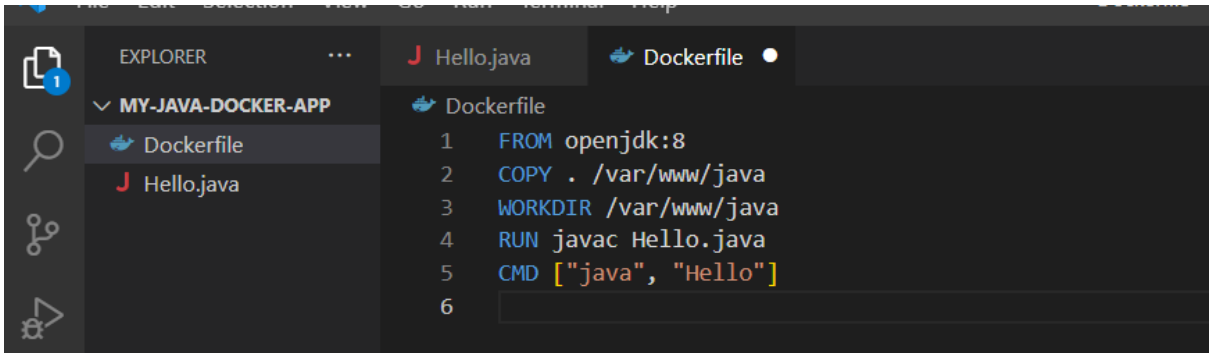
Syntax: cd directory_name

```
C:\Users\Aakash007>mkdir my-java-docker-app

C:\Users\Aakash007>cd my-java-docker-app

C:\Users\Aakash007\my-java-docker-app>code .
```

Now create a Java file. Save this file as Hello.java file.

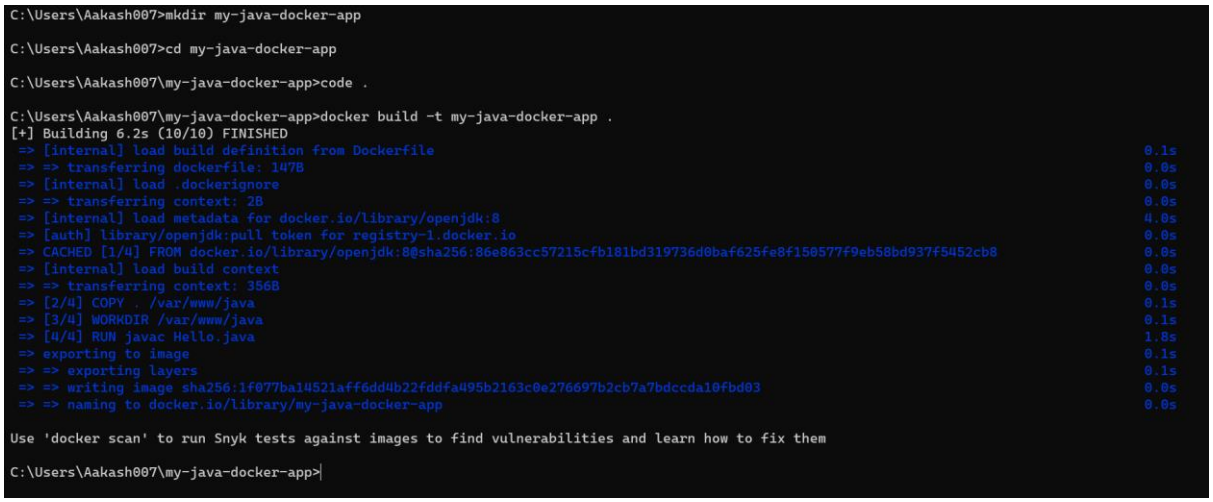


Now create a docker file with name “Dockerfile”. This file does not contains any extensions. It will contains the instructions for the docker.



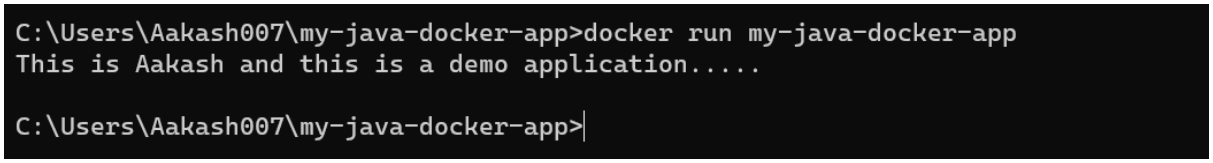
Now create an image by using the following command.

Syntax: docker build -t app_name .



Now run the app by using the following command.

Syntax: docker run app_name



WE HAVE SUCCESSFULLY CREATED OUR JAVA APPLICATION IN DOCKER.