Fall 2023

Lab Assignment #5:

Due Dates	
Post-Lab	**Submit before 23:59 on Friday , November 3, 2023**

The objectives of this lab are to:

to get you familiar with docker and docker hub

The following rules apply to this lab and all other lab assignments in the future:

1. Before submitting your lab reports, take a moment to make sure that you are handing in all the material that is required. If you forget to hand something in, that is your fault; you can't use `I forgot' as an excuse to hand in parts of the assignment late.
2. **20% marks** will be deducted from the assignments handed in up to **24 hours** after each due date. It means if your mark is X out of Y, you will only gain 0.8 times X. There will be no credit for assignments turned in later than 24 hours after the due dates; they will be returned unmarked.

In Lab (30 marks)

In this assignment, you will get acquainted with the concept of containers and their use. We will use Docker for this.

Containers are actually mini microservices. Consider the attached Python script testprint.py to be the logic of the microservice.

You are to create a docker container that can run the attached Python script and output the following text: ” This is a container test”.

Please document with your execution steps with screen prints and text to demonstrate that you completed the exercise successfully and hand in this document for grading.

Task - 1: Detail of Documentation (20 Marks)

Task - 2: Parameter Research (10 Marks)

Python script

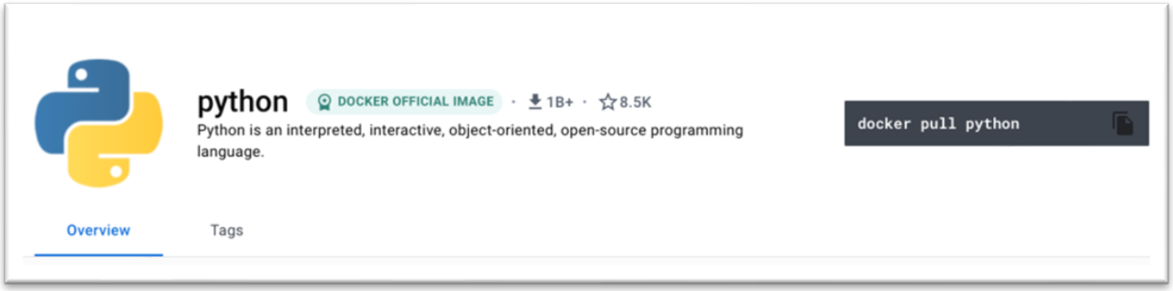
1. **Install a docker desktop on your computer.**
   https://docs.docker.com/engine/install/

   Once installed make sure docker is running.

2. **Create a Docker Hub account.**
   https://hub.docker.com

   and sign on.

3. **Find a Python image that you could use on the docker hub. The following is a recommendation.**



Python is an interpreted, interactive, object-oriented, open-source programming language.

```
docker pull python
```

4. **Now go to the cmd line on your computer and run the following command.**

   ```
   Docker pull python
   ```

   This will pull the image from the docker hub to your docker environment.

5. **Once done you can run the following command:**

   ```
   Docker image ls
   ```

   This will give you all the images that are on your computer. You should also see the Python image you just downloaded.

6. **Now rename this image to something more useful.**

   ```
   Docker tag python ensf607
   ```

   This will rename your image to ensf607. You can use another name if you wish.

7. **Now run a container using this image within the docker**

   ```
   Docker run -itd ensf607
   ```

   Explore what the options **-itd** mean and are used for. **Please include the description of these options in your document**

8. **Now you should see all the active containers that are running with.**

   ```
   Docker ps
   ```

9. **You will see that docker assigns a random name (id far right) to your loaded image**.

10. **Rename the Python container.**

    ```
    Docker rename <name of running container> < new name>
    ```

Do another docker ps

You should see your container with the new name you assigned.

11. **Now we want to go into the container. This is a linux container so you need to use linux commands**

12. **To enter the container run the command.**

```
Docker exec -it <your containername> sh
```

sh stands for shell environment or shell commands. This allows you to run Linux commands inside the container.

Once you are in your container try these Linux commands

Ls -ltr ➔ list directories and files

Go to the home directory with

```
cd ./home
```

Within the home directory create a new directory called python_scripts

```
mkdir ./python_scripts
```

Moving up the directory tree use

```
cd ../
```

To use the current directory use ./

List all files and directories.

```
ls -ltr
```

Remove a directory

```
rm ./<directory name or file>
```

Use the exit to get out of your container and back to your computer OS. Command line

13. **Now we want to upload our little Python script named testprint.py.**

14. **Run the following command to upload the python script.**

```
docker cp <your folder structure on your computer> \testprint.py  <name of your container>:/home/python_scripts/testprint.py
```

Now go back into your container with

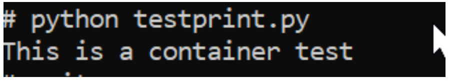
```
Docker exec -it <your containername> sh
```

Go to the ./home/python_script directory and do a ls -ltr you should see the python script. testprint.py

15. **Run the command:**

```
python ./testprint.py
```

And you should see the following output:



```
# python testprint.py
This is a container test
```

You have now successfully executed a Python script in a container.
CONGRATULATIONS.

We could now also install a web server or use an image with a web service to deploy web applications to this container and access this application via our browser. But that is for your own enjoyment.

You could also upload this container into your docker hub repository to download it onto another machine i.e cloud VM or other platforms. Or you could also provide this container for others to use. Lots of great possibilities