ECOLE NATIONALE
SUPÉRIEURE
D'INFORMATIQUE

الجـــمهوريـــة الجزائريـــة الديمـــقراطيـــــة الشـعبيـة
وزارة التعليـــم العـــــالي والبـــحث العلمـــي
People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research

Virtualization and Cloud Computing, VCL

2nd Year Specialty SIQ G02, 2CS SIQ2

LAB5A Report

Containerization with Docker

Studied by:

HEDDADJI Nour El Imane

SOLTANI MERIEM

E-mails :

jn_heddadji@esi.dz

jm_soltani@esi.dz

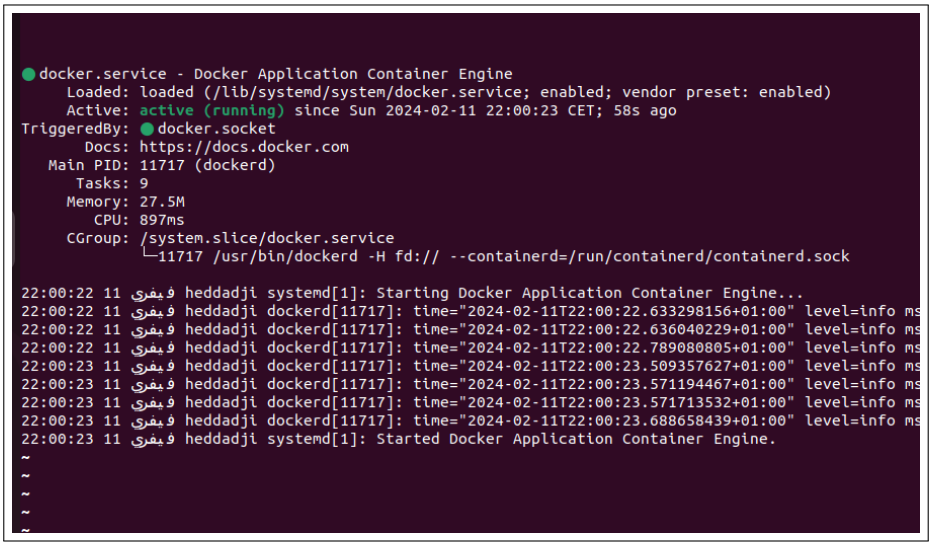
Contents

1. Docker Installation

To proceed with this lab, we created the following:

- An Ubuntu server.
- A Docker Hub account

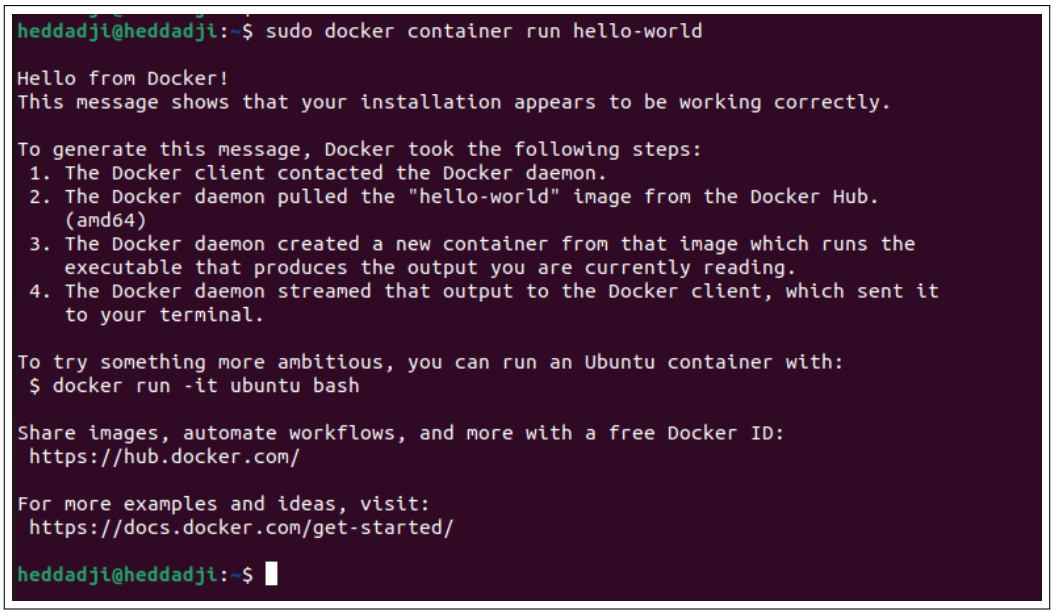
```

    sudo apt update
    sudo apt install apt-transport-https ca-certificates curl software-properties-common
    curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
    sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu bionic stable"
    sudo apt install docker-ce
    #Verify
    sudo systemctl status docker
.
```



2. Getting familiar with Docker

2.1. Hello-world Image

```

    docker container run hello-world
.
```



2.2. Search for an Image in the Registry

This command searches the Docker Hub registry for images related to ”example”.

It returns a list of images available on the registry that match the search term.

```

    docker search example
.
```

2.3. Pulling an image

Pulling an image means **downloading** it to your local machine, making it available for local use.

```

    docker image pull image_name
.
```

2.4. Listing Local Docker Images

```

    docker image ls
.
```

```
heddadji@heddadji:~$ sudo docker search ns2
[sudo] password for heddadji:
NAME                                      DESCRIPTION                                     STARS     OFFICIAL   AUTOMATED
ns208/guestbook-php-frontend                                                              0
tikz/ns2-roundend                                                                         0
tikz/ns2-skill-go                                                                         0
tikz/ns2-discord                                                                          0
tikz/ns2sud-web                                                                           0
ns208/centos6-sshd                                                                        0
tikz/ns2-web                                                                              0
ns208/mycentos-ssh                                                                        0
ekiourk/ns2                               Network Simulator 2.35                          3
tikz/ns2-skill                                                                            0
zenithal/ns2discordbot                                                                    0
massimilianomirelli/ns2_server                                                            0
ns29/get-started                                                                          0
serverboi/ns2                                                                             0
ns265422/julia                                                                            0
ns2808/cheers2019                                                                         0
ns29/cloudserver                                                                          0
christofferbraun/ns2-jenkins-postgresql                                                   0
npranav10/ns2                                                                             0
esion/ns2_server-docker                   Docker Image for Natural Selection 2 Server     0
dotrdg/ns2                                                                                0
kersey/ns2.35                             Comes with GNUPlot                              0
jangelinav/ns2                                                                            0
gcasella/ns2-docker                                                                       0
ns2rk/docker101tutorial                                                                   0
heddadji@heddadji:~$ sudo docker image pull ekiourk/ns2
```

```
heddadji@heddadji:~$ sudo docker image pull ekiourk/ns2
Using default tag: latest
latest: Pulling from ekiourk/ns2
Image docker.io/ekiourk/ns2:latest uses outdated schema1 manifest format. Please upgrade to a schema2 image for better
ibility. More information at https://docs.docker.com/registry/spec/deprecated-schema-v1/
a3ed95caeb02: Pull complete
2d7928839f96: Pull complete
2cbe3e3fa432: Pull complete
76a59787cc98: Pull complete
2eb92b26d2c9: Pull complete
0375c530f5ff: Pull complete
d5da3d004d16: Pull complete
cdfbbc36a11a: Pull complete
Digest: sha256:8c2764a3f35baa783af4dde92c257c6221b8f45279ccf1e84ca21c526df26ef1
Status: Downloaded newer image for ekiourk/ns2:latest
docker.io/ekiourk/ns2:latest
heddadji@heddadji:~$ sudo  docker image ls
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
hello-world   latest    d2c94e258dcb   9 months ago   13.3kB
ekiourk/ns2   latest    afe271929563   8 years ago    743MB
heddadji@heddadji:~$
```

2.5. Docker groups

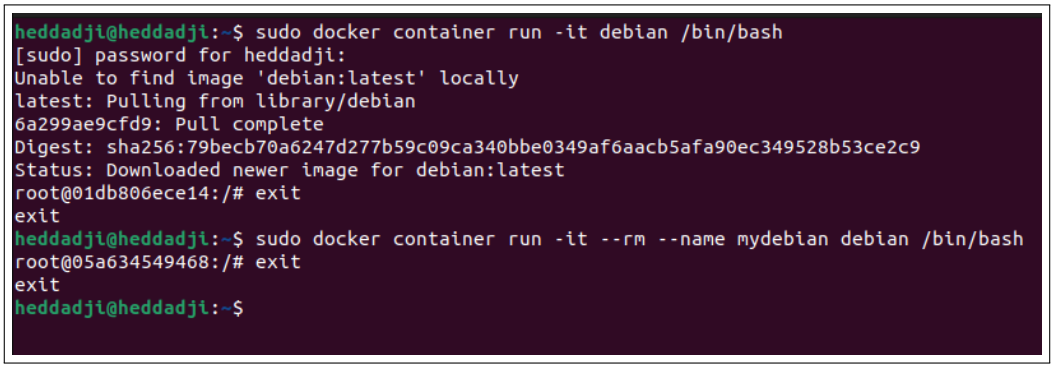
To avoid using sudo each time we run a Docker command.

```

    sudo usermod -aG docker your_username
    groups
.
```

3. Managing Containers

```

    # Start an interactive Debian container and launch a Bash shell
    docker container run -it debian /bin/bash

    #  Start an interactive Debian container with automatic removal and assign a custom name
    docker container run -it --rm --name mydebian debian /bin/bash


```

```
.
```



4. Custom Docker images

A Dockerfile is a text file that contains instructions for creating a Docker image. Instructions are written in the format: **DIRECTIVE argument**.

4.1. Dockerfile Structure

```

    #Create the folder for the project
    mkdir app
    # Create dockerfile
    touch Dockerfile
    nano Dockerfile
    # Opens the Dockerfile in the nano text editor
.
```

This is how a dockerfile should look like :

```

   # Specify the base image
    FROM your_base_image:tag

    # Set the working directory
    WORKDIR /path_to_working_directory

    # Install dependencies
    RUN your_package_manager install your_dependencies

    # Copy application files to the container
    COPY . /app

    # Specify environment variables
```

```
    ENV YOUR_ENV_VARIABLE=value

    # Define the default command to run when the container starts
    CMD ["your_command", "--your-option=option_value"]

.
```

4.2. Building and Running the Custom Image

Open a terminal and navigate to the directory containing the Dockerfile.

```

    # Build the Docker image
    docker build . -t my-custom-image:v1

    # Check the list of Docker images
    docker image ls

    # Run the Docker container
    docker container run -d my-custom-image :v1

.
```

5. Hosting a Python/Flask app on a docker

We will create these files :

- app.py : to store the code for our flask app
- Dockerfile

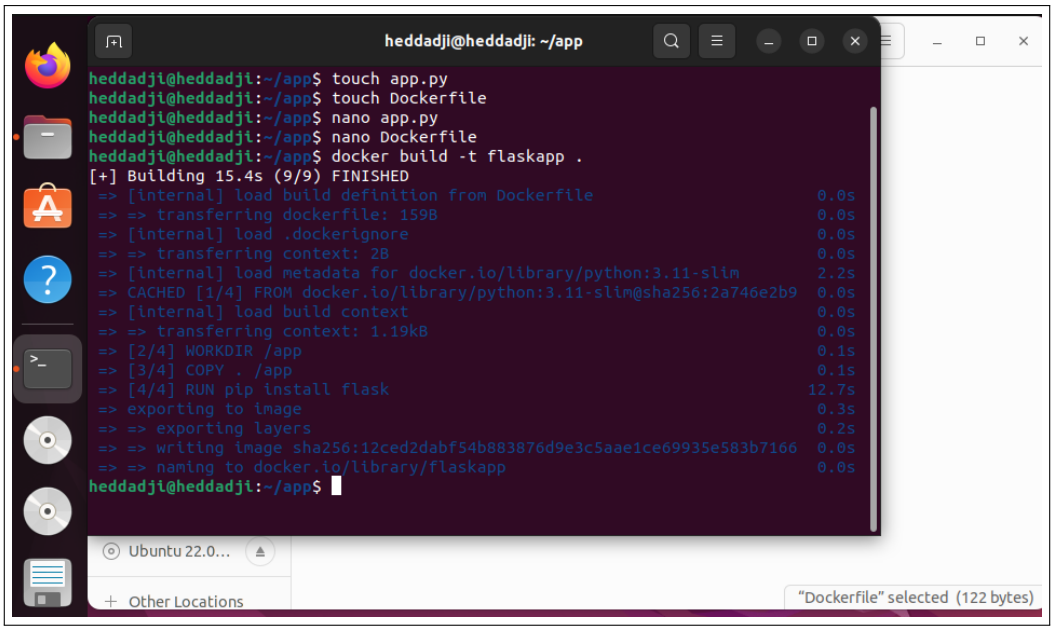
Flask file : app.py

```

    from flask import Flask, render_template_string

    app = Flask(__name__)

    @app.route('/')
    def hello_world():
        html_content = """
        <!DOCTYPE html>
        <html lang="en">
        <head>
            <meta charset="UTF-8">
            <meta http-equiv="X-UA-Compatible" content="IE=edge">
```

```
            <meta name="viewport" content="width=device-width, initial-scale=1.0">
            <title>Mon Premier Conteneur</title>
            <style>
                body {
                    font-family: Arial, sans-serif;
                    background-color: #f4f4f4;
                    text-align: center;
                    margin: 100px;
                }
                h1 {
                    color: #333;
                }
            </style>
        </head>
        <body>
            <h1>Mon Premier Conteneur</h1>
            <p>Test successful!</p>
            <p> Done by : Heddadji and Soltani SIQ2 VCL module </p>
        </body>
        </html>
        """
        return render_template_string(html_content)

    if __name__ == '__main__':
        app.run(host='0.0.0.0', port=8081)

.
```

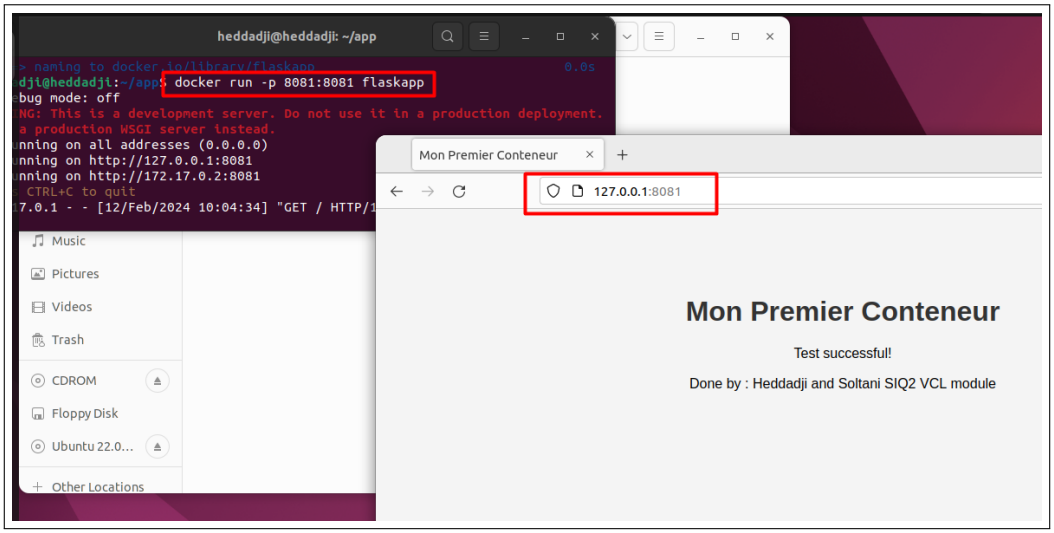
Dockerfile

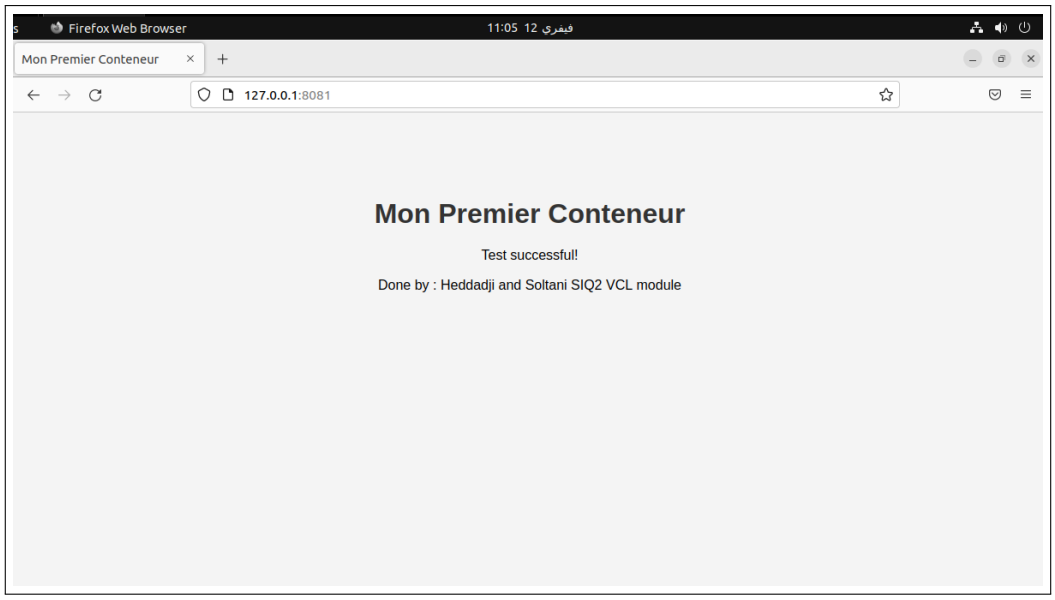
```

    FROM python:3.11-slim

    WORKDIR /app

    COPY . /app

    RUN pip install flask

    CMD ["flask", "run", "--host=0.0.0.0", "--port=8081"]

.
```



5.1. Building and Running the Docker Image

```
cd lab
// . : current dir
docker build -t flaskapp .
docker run -p 8081:8081 flaskapp
#Acess in http://localhost:8081
.
```





5.2. Hosting the Docker image to Docker Hub

1. **Create a Docker Hub Account:** `https://hub.docker.com/`.
2. **Login to Docker Hub:**

```
docker login

```

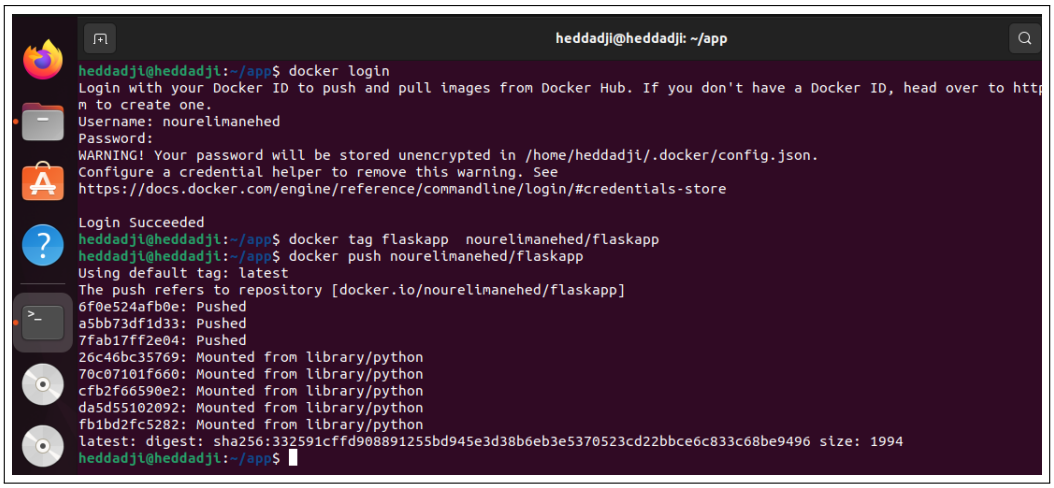
   Enter your Docker Hub username and password when prompted.

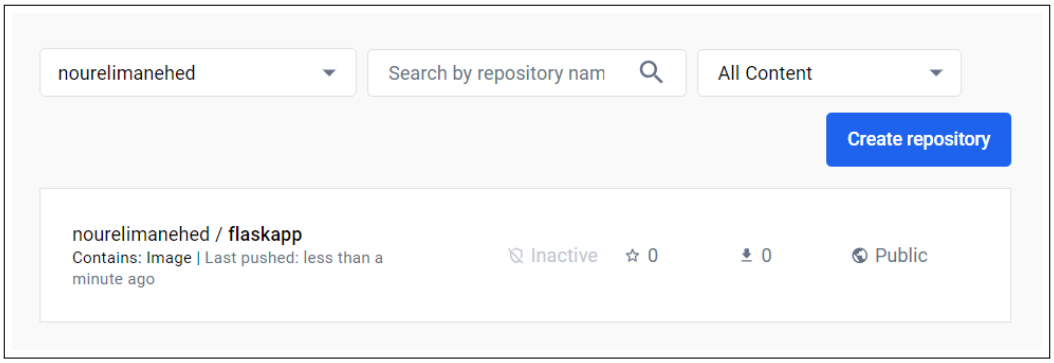
3. **Tag your Docker Image:** with Docker Hub username and the desired repository name.

```
docker tag flaskapp your-username/repository-name

```

4. **Push the Docker Image:**

```
docker push your-username/repository-name

```


heddadji@heddadji: ~/app
heddadji@heddadji:~/app$ 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 http
m to create one.
Username: nourelimanehed
Password:
WARNING! Your password will be stored unencrypted in /home/heddadji/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store
Login Succeeded
heddadji@heddadji:~/app$ docker tag flaskapp nourelimanehed/flaskapp
heddadji@heddadji:~/app$ docker push nourelimanehed/flaskapp
Using default tag: latest
The push refers to repository [docker.io/nourelimanehed/flaskapp]
6f0e524afb0e: Pushed
a5bb73df1d33: Pushed
7fab17ff2e04: Pushed
26c46bc35769: Mounted from library/python
70c07101f660: Mounted from library/python
cfb2f66590e2: Mounted from library/python
da5d55102092: Mounted from library/python
fb1bd2fc5282: Mounted from library/python
latest: digest: sha256:332591cffd908891255bd945e3d38b6eb3e5370523cd22bbce6c833c68be9496 size: 1994
heddadji@heddadji:~/app$


nourelimanehed
Search by repository nam
All Content
Create repository
nourelimanehed / flaskapp
Contains: Image | Last pushed: less than a minute ago
Inactive
0
0
Public