ECOLE NORMALE SUPÉRIEURE DE
L'ENSEIGNEMENT TECHNIQUE DE MOHAMMEDIA

UNIVERSITÉ HASSAN II DE CASABLANCA

ENSET

المدرسة العليا لأساتذة التعليم التقني
المحمدية
جامعة الحسن الثاني بالدار البيضاء

DEPARTEMENT MATHEMATIQUES ET INFORMATIQUE

Compte Rendu

Filière :
« Ingénierie Informatique : Big Data et Cloud Computing »

II-BDCC

TP 2 : Docker (Installation, Gestion de Conteneurs et d'Images, Docker Hub, APIREST Docker, Création et Exécution d'une Image Docker, Plugin Eclipse)

Réalisé par :
Braimi Hamza

Encadré par :
Pr. Kamal EL GUEMMAT

Année Universitaire : 2023-2024

Table of Contents

Introduction OF PW 2: Docker

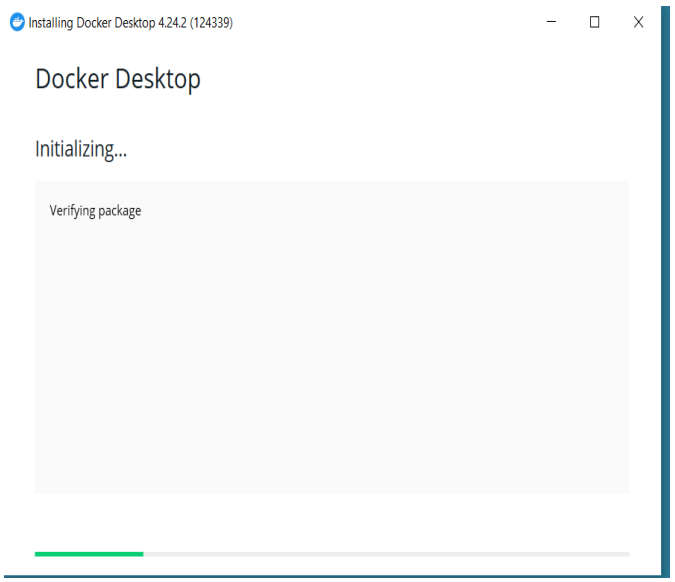
In this practical guide, we will walk through the essential steps to work with Docker, a leading containerization platform. Docker allows us to package applications and their dependencies into lightweight containers, making it easier to develop, deploy, and manage applications consistently across different environments. Whether we are running Windows or Ubuntu, we will learn how to install Docker, manage containers and images, and leverage Docker Hub, a repository for Docker images.

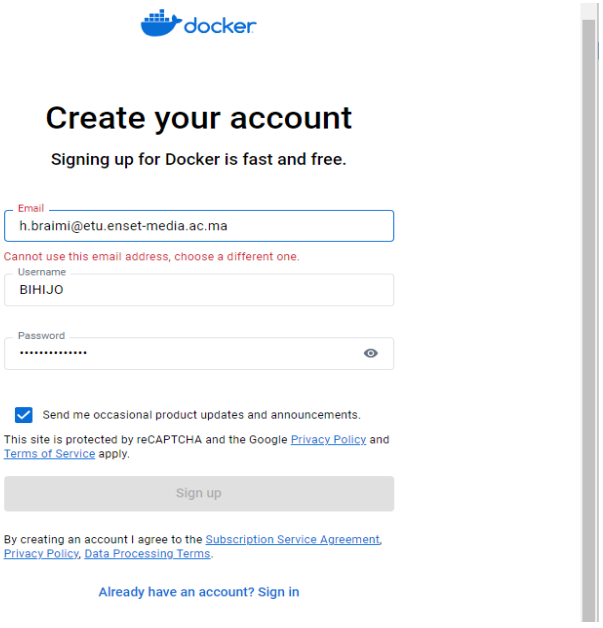
Part 1 Docker Installation

1. On Windows:

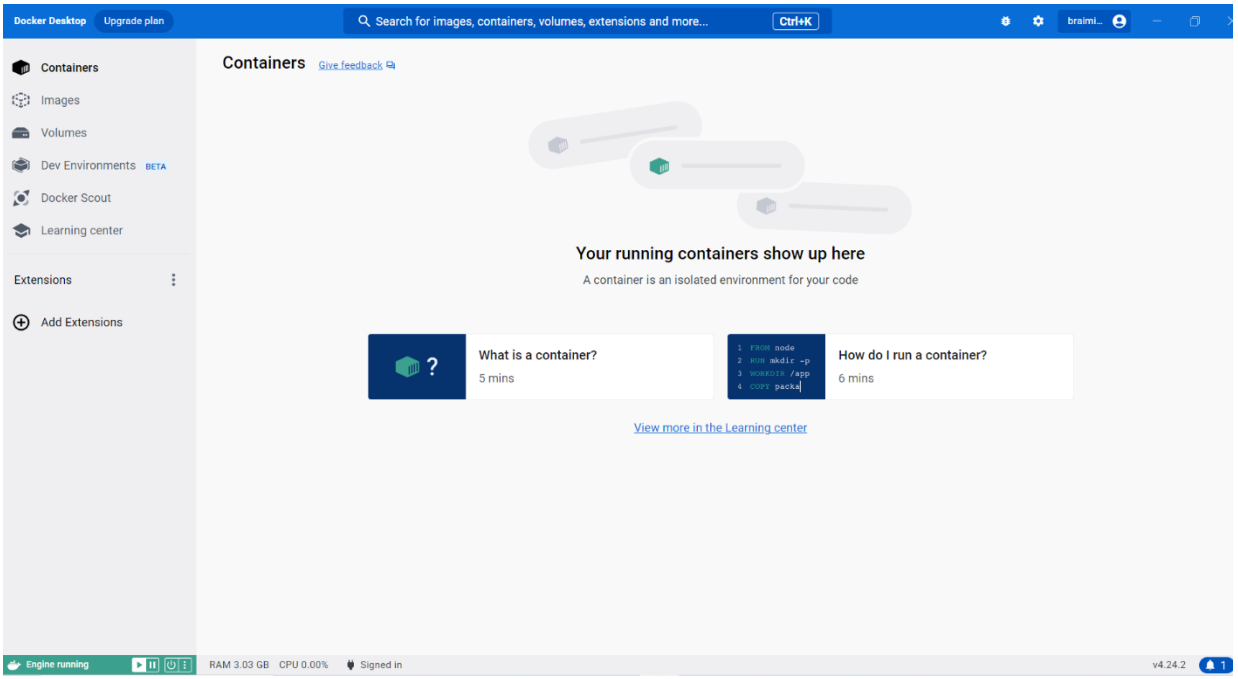
Download Docker Desktop from the official website: Docker Desktop for Windows

Run the installer with the default settings.





- Lastly, we get this awesome workspace.



PART 2: Container and Image Management, Docker Hub

1. **Show Docker version:**

   Using this command, we will get the version of Docker **"docker --version"**

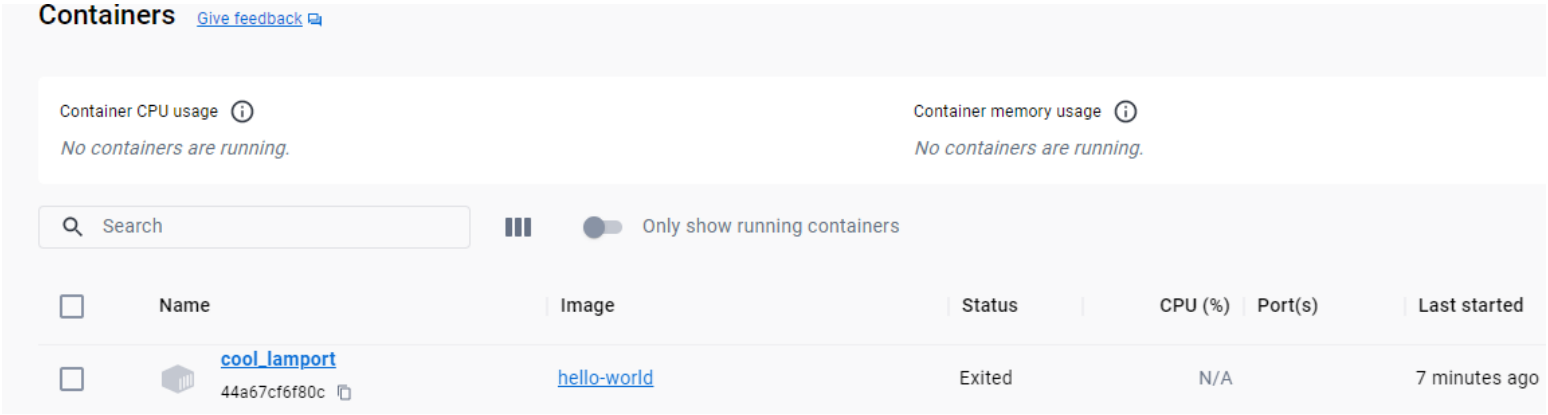
2. **Run the hello-world and whale say images from Docker Hub.**

   To run the "hello-world" and "whale say" images from Docker Hub, execute the following commands:

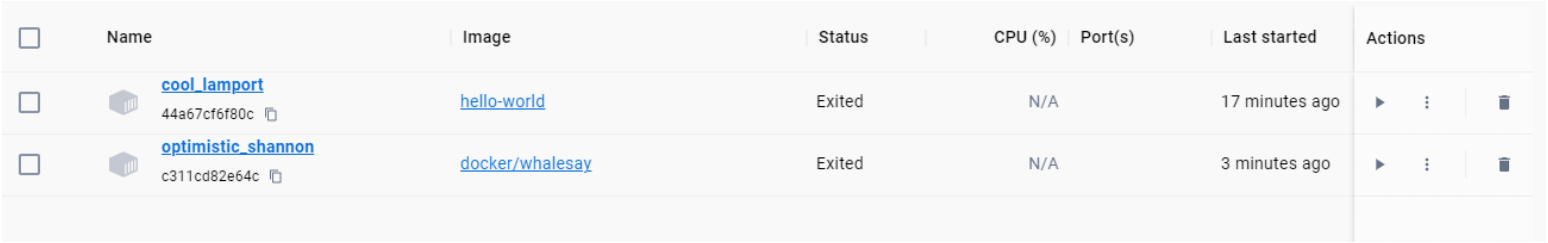
   - docker run hello-world
   - docker run docker/whalesay cowsay "Hello, BDCC!"

```
C:\Users\HAMZA>docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
719385e32844: Pull complete
Digest: sha256:88ec0acaa3ec199d3b7eaf73588f4518c25f9d34f58ce9a0df68429c5af48e8d
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/
```

So if we check now our workspace we will find that new image has been created




```
C:\Users\HAMZA> docker run docker/whalesay cowsay "Hello, BDCC!"
Unable to find image 'docker/whalesay:latest' locally
latest: Pulling from docker/whalesay
[DEPRECATION NOTICE] Docker Image Format v1, and Docker Image manifest version 2, schema 1 support will be removed in an upcoming release. Suggest the author of docker.io/docker/whalesay:latest to upgrade the
age to the OCI Format, or Docker Image manifest v2, schema 2. More information at https://docs.docker.com/go/deprecated-image-specs/
e190868d63f8: Pull complete
909cd34c6fd7: Pull complete
0b9bfabab7c1: Pull complete
a3ed95caeb02: Pull complete
00bf65475aba: Pull complete
c57b6bcc83e3: Pull complete
8978f6879e2f: Pull complete
8eed3712d2cf: Pull complete
Digest: sha256:178598e51a26abbc958b8a2e48825c90bc22e641de3d31e18aaf55f3258ba93b
Status: Downloaded newer image for docker/whalesay:latest
 _____
< Hello, BDCC! >
 -------------
    \
     \
      \
                    ##        .
              ## ## ##       ==
           ## ## ## ##      ===
       /""""""""""""""""___/ ===
  ~~~~ {~~ ~~~~ ~~~~ ~~~~ ~~ ~ /  ===- ~~~
       \______ o          __/
        \    \        __/
          \____\______/
```



In summary, the first command (docker run hello-world) is used to check if Docker is correctly installed and running, while the second command (docker run docker/whalesay cowsay "Hello, BDCC!") is more of a fun and whimsical demonstration of how you can run containers with custom commands and images in Docker.

3. List active containers

To list active containers, you can use several commands:

- To list all containers (including stopped ones): **docker ps -a**

```
C:\Users\HAMZA>docker ps -a
CONTAINER ID   IMAGE             COMMAND                  CREATED          STATUS                      PORTS     NAMES
c311cd82e64c   docker/whalesay   "cowsay 'Hello, BDCC…"   8 minutes ago    Exited (0) 7 minutes ago              optimistic_shannon
44a67cf6f80c   hello-world       "/hello"                 21 minutes ago   Exited (0) 21 minutes ago             cool_lamport
```

- To list running containers: **docker ps**

```
C:\Users\HAMZA>docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
```

4. Remove the "hello-world" and "whalesay" images.

To remove the "hello-world" and "whalesay" images and containers, you can use these commands:

- docker stop <container_id> # Stop the containers
- docker rm <container_id> # Remove the containers
- docker rmi hello-world docker/whalesay # Remove the images

```
C:\Users\HAMZA>docker rm c311cd82e64c
c311cd82e64c
```

```
C:\Users\HAMZA>docker rmi hello-world -f
Untagged: hello-world:latest
Untagged: hello-world@sha256:88ec0acaa3ec199d3b7eaf73588f4518c25f9d34f58ce9a0df68429c5af48e8d
Deleted: sha256:9c7a54a9a43cca047013b82af109fe963fde787f63f9e016fdc3384500c2823d
```

```
C:\Users\HAMZA>docker rm 44a67cf6f80c
44a67cf6f80c

C:\Users\HAMZA>docker rmi docker/whalesay -f
Untagged: docker/whalesay:latest
Untagged: docker/whalesay@sha256:178598e51a26abbc958b8a2e48825c90bc22e641de3d31e18aaf55f3258ba93b
Deleted: sha256:6b362a9f73eb8c33b48c95f4fcce1b6637fc25646728cf7fb0679b2da273c3f4
Deleted: sha256:34dd66b3cb4467517d0c5c7dbe320b84539fbb58bc21702d2f749a5c932b3a38
Deleted: sha256:52f57e48814ed1bb08a651ef7f91f191db3680212a96b7f318bff0904fed2e65
Deleted: sha256:72915b616c0db6345e52a2c536de38e29208d945889eecef01d0fef0ed207ce8
Deleted: sha256:4ee0c1e90444c9b56880381aff6455f149c92c9a29c3774919632ded4f728d6b
Deleted: sha256:86ac1c0970bf5ea1bf482edb0ba83dbc88fefb1ac431d3020f134691d749d9a6
Deleted: sha256:5c4ac45a28f91f851b66af332a452cba25bd74a811f7e3884ed8723570ad6bc8
Deleted: sha256:088f9eb16f16713e449903f7edb4016084de8234d73a45b1882cf29b1f753a5a
Deleted: sha256:799115b9fdd1511e8af8a8a3c8b450d81aa842bbf3c9f88e9126d264b232c598
Deleted: sha256:3549adbf614379d5c33ef0c5c6486a0d3f577ba3341f573be91b4ba1d8c60ce4
Deleted: sha256:1154ba695078d29ea6c4e1adb55c463959cd77509adf09710e2315827d66271a
```

5. **List active containers**

```
C:\Users\HAMZA>docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
```

6. **Run a database server image in the background.**

To run a database server image in the background with a password, use the -d flag for detaching and the -e flag to set an environment variable for the password:

- **docker run -d -e PMA_password=<your_password> <database_image>**

Pull the Docker Image We need to pull the MySQL image from Docker Hub. Run the following command to download the image:

- **docker pull MySQL**

```
C:\Users\HAMZA>docker pull mysql
Using default tag: latest
latest: Pulling from library/mysql
8e0176adc18c: Pull complete
14e977b0f4b2: Pull complete
a7b58dd6f78b: Pull complete
fba70cc872a5: Pull complete
5db2cc6eab8f: Pull complete
081f41f573ba: Pull complete
86bf2dc4ded9: Pull complete
47f08b0e916e: Pull complete
850e29ae8eeb: Pull complete
13517fe0d921: Pull complete
Digest: sha256:f61944ff3f2961363a4d22913b2ac581523273679d7e14dd26e8db8c9f571a7e
Status: Downloaded newer image for mysql:latest
docker.io/library/mysql:latest

What's Next?
  View a summary of image vulnerabilities and recommendations → docker scout quickview mysql
```

Run the Container in the Background: To run the MySQL server container in the background with the -d option, use the following command:

- **docker run -d --name mysql-server -e MYSQL_ROOT_PASSWORD=your_password mysql**

```
C:\Users\HAMZA>docker run -d --name my_database_container -e MYSQL_ROOT_PASSWORD=your_password mysql
4614d7fd1a8f559851df4271e0a364b0d292675a14f7b8b28309c8bd6521944f
```

7. **Run an administration image for the database.**

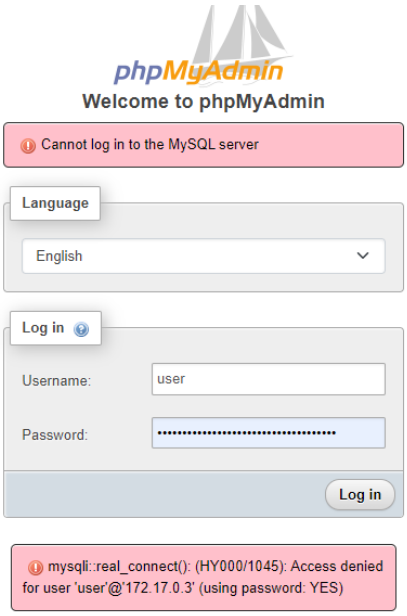
```
C:\Users\HAMZA>docker run -d --name mysql-server -e MYSQL_ROOT_PASSWORD=your_password mysql
5e04d1c2cddf70093af036a880168c29521899ec6ba14744aac29983fbd14790

C:\Users\HAMZA>docker run -d --name phpmyadmin-container --link mysql-server:db -e PMA_HOST=db -p 8080:80 phpmyadmin/phpmyadmin
Unable to find image 'phpmyadmin/phpmyadmin:latest' locally
latest: Pulling from phpmyadmin/phpmyadmin
faef57eae888: Pull complete
989a1d6c052e: Pull complete
0705c9c2f22d: Pull complete
621478e043ce: Pull complete
98246dcca987: Pull complete
bfed8c155cb6: Pull complete
7a7c2e908867: Pull complete
d176994b625c: Pull complete
2d8ace6a2716: Pull complete
c70df516383c: Pull complete
15e1b44fe4c7: Pull complete
65e50d44e95a: Pull complete
77f68910bc0a: Pull complete
605dd3a6e332: Pull complete
99ce27188f07: Pull complete
74d64e32c5d5: Pull complete
ef5fc9928b9f: Pull complete
163f3256e112: Pull complete
Digest: sha256:67ba2550fd004399ab0b95b64021a88ea544011e566a9a1995180a3decb6410d
Status: Downloaded newer image for phpmyadmin/phpmyadmin:latest
21406d4093c662756227d5b110d53c44c8f85c38f75b429955a5427b0b9ab80c
```

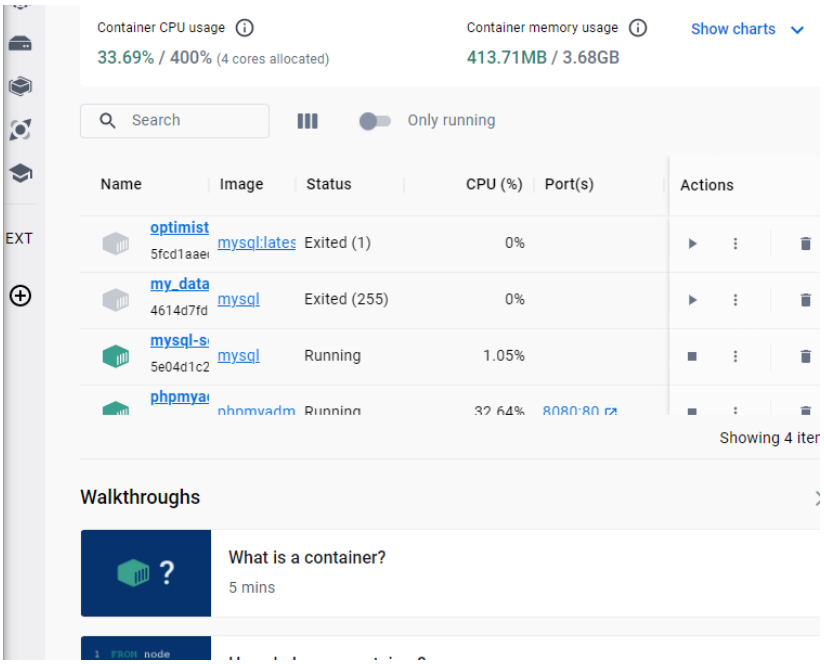
Make sure your MySQL server container is running. You can use the command you provided earlier to start the MySQL server:

- **docker run -d --name mysql-server -e MYSQL_ROOT_PASSWORD=your_password mysql**

Now, you can run the phpMyAdmin container and link it to the MySQL server container using the --link parameter:

- **docker run -d --name phpmyadmin-container --link mysql-server:db -e PMA_HOST=db -p 8080:80 phpmyadmin/phpMyAdmin**

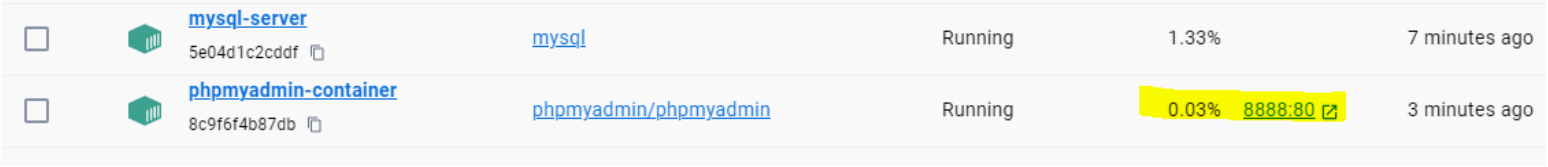




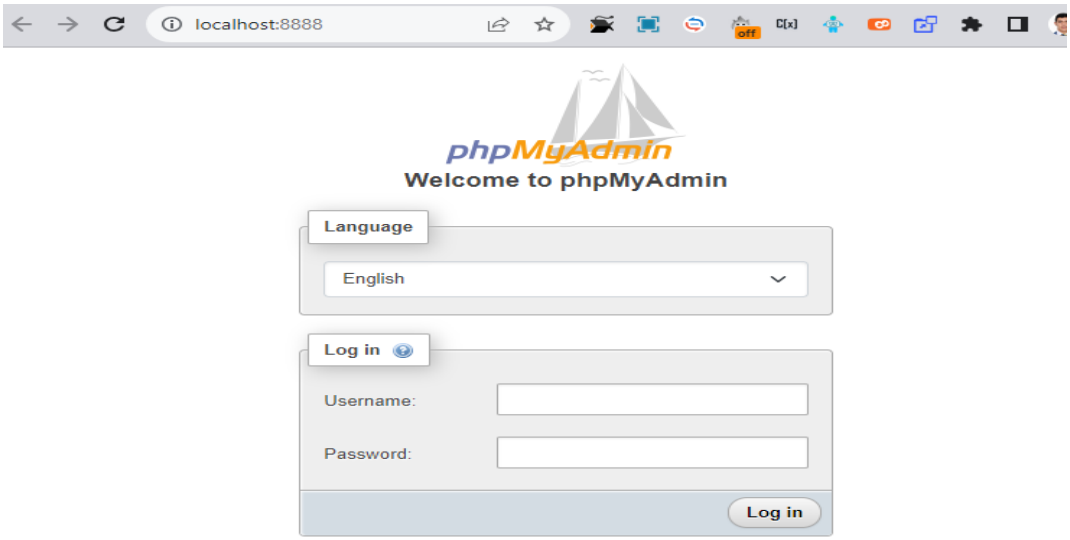
8. Change the port of the administration container.

To change the port of the phpMyAdmin container by mapping ports, you can modify the docker run command as follows. Suppose we want to change the phpMyAdmin container's external port to 8888 while keeping its internal port as 80 (the default for web servers). Here's the updated command:

```
C:\Users\HAMZA>docker stop phpmyadmin-container
phpmyadmin-container

C:\Users\HAMZA>docker rm phpmyadmin-container
phpmyadmin-container

C:\Users\HAMZA>docker run -d --name phpmyadmin-container --link mysql-server:db -e PMA_HOST=db -p 8888:80 phpmyadmin/phpmyadmin
8c9f6f4b87db2b1fd00139642c5964caad3fe87600668ff77a6d72a1572d3ed5
```



9. Access the phpMyAdmin container.



10. Enter interactive mode on the MySQL server

To enter interactive mode on the MySQL server container and execute some commands, you can use the docker exec -it command. Here's how to do it:

Assuming you have a running MySQL server container with the name "mysql-server," you can enter interactive mode as follows:

```
C:\Users\HAMZA>docker exec -it mysql-server bash
bash-4.4# sdf
bash: sdf: command not found
bash-4.4# sdf
bash: sdf: command not found
bash-4.4# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.1.0 MySQL Community Server - GPL

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

11. Establish a connection with the MySQL server container

Assuming we have a running MySQL server container with the name "mysql-server," we can use the docker attach command to connect to this container. However, please note that this command will attach to the main process of the container (usually the command shell), and you won't be in an interactive MySQL environment.

```
C:\Users\HAMZA>docker attach mysql-server
2023-10-21T21:07:48.586585Z 1 [System] [MY-013577] [InnoDB] InnoDB initialization has ended.
2023-10-21T21:07:50.512818Z 0 [Warning] [MY-010068] [Server] CA certificate ca.pem is self signed.
2023-10-21T21:07:50.512911Z 0 [System] [MY-013602] [Server] Channel mysql_main configured to support TLS. Encrypted conn
ections are now supported for this channel.
2023-10-21T21:07:50.543724Z 0 [Warning] [MY-011810] [Server] Insecure configuration for --pid-file: Location '/var/run/m
ysqld' in the path is accessible to all OS users. Consider choosing a different directory.
2023-10-21T21:07:50.691438Z 0 [System] [MY-010931] [Server] /usr/sbin/mysqld: ready for connections. Version: '8.1.0'  s
ocket: '/var/run/mysqld/mysqld.sock'  port: 3306  MySQL Community Server - GPL.
2023-10-21T21:07:50.691520Z 0 [System] [MY-011323] [Server] X Plugin ready for connections. Bind-address: '::' port: 330
60, socket: /var/run/mysqld/mysqlx.sock
```

12. Inspect Containers

```
C:\Users\HAMZA>docker inspect mysql-server
[
    {
        "Id": "5e04d1c2cddf70093af036a880168c29521899ec6ba14744aac29983fbd14790",
        "Created": "2023-10-21T19:20:02.804695888Z",
        "Path": "docker-entrypoint.sh",
        "Args": [
            "mysqld"
        ],
        "State": {
            "Status": "running",
            "Running": true,
            "Paused": false,
            "Restarting": false,
            "OOMKilled": false,
            "Dead": false,
            "Pid": 16765,
            "ExitCode": 0,
            "Error": "",
            "StartedAt": "2023-10-21T21:11:12.304973378Z",
            "FinishedAt": "2023-10-21T21:11:07.044115762"
        },
        "Image": "sha256:ae2502152260b33bfafb9c3e3a1811086317e0236abf83a59503cec0f8980573",
        "ResolvConfPath": "/var/lib/docker/containers/5e04d1c2cddf70093af036a880168c29521899ec6ba14744aac29983fbd14790/resolv.conf",
        "HostnamePath": "/var/lib/docker/containers/5e04d1c2cddf70093af036a880168c29521899ec6ba14744aac29983fbd14790/hostname",
        "HostsPath": "/var/lib/docker/containers/5e04d1c2cddf70093af036a880168c29521899ec6ba14744aac29983fbd14790/hosts",
        "LogPath": "/var/lib/docker/containers/5e04d1c2cddf70093af036a880168c29521899ec6ba14744aac29983fbd14790/5e04d1c2cddf70093af036a880168c29521899ec6ba14744aac29983fbd14790-j
        "Name": "/mysql-server",
        "RestartCount": 0,
        "Driver": "overlay2",
        "Platform": "linux",
        "MountLabel": "",
        "ProcessLabel": "",
        "AppArmorProfile": "",
        "ExecIDs": null,
        "HostConfig": {
            "Binds": null,
            "ContainerIDFile": "",
            "LogConfig": {
                "Type": "json-file",
                "Config": {}
            },
            "NetworkMode": "default",
            "PortBindings": {},
            "RestartPolicy": {
                "Name": "no",
```

13. Stop And Start Containers

To stop a running container, use the docker stop command, and to start a stopped container, use the docker start command.

To stop a container:

- **Docker stop my-container.**

To start a container:

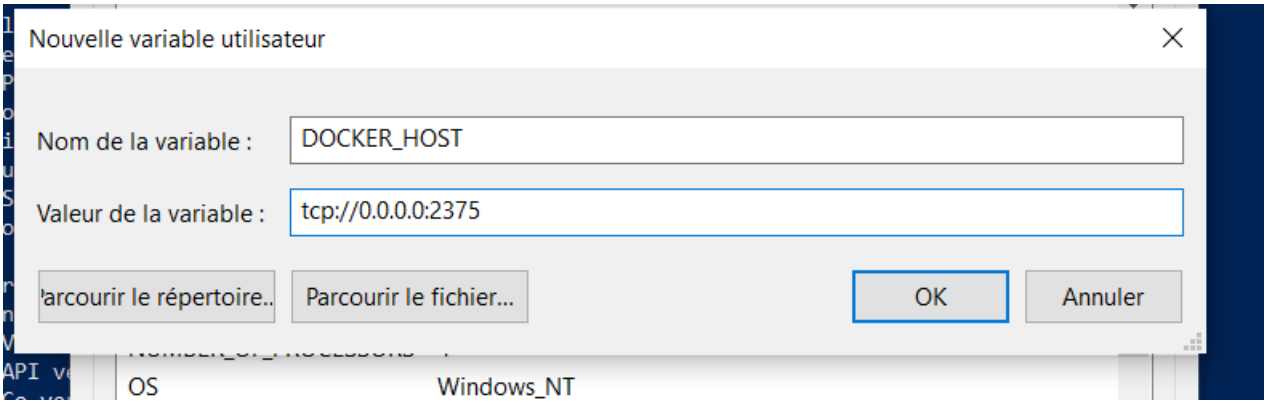
- **Docker start my-container.**

```
C:\Users\HAMZA>docker stop mysql-server
mysql-server

C:\Users\HAMZA>docker start mysql-server
mysql-server
```

PART 3: APIREST Docker, Creating and Running a Docker Image, Eclipse Plugin

1. **Activate the Docker REST API and test it remotely with Docker CLI:**

```
PS C:\Users\HAMZA> Set-ExecutionPolicy Bypass -Scope Process -Force ;
PS C:\Users\HAMZA> iex((New-Object System.Net.WebClient).DownloadString('https://chocolatey.org/install.ps1'))
Forcing web requests to allow TLS v1.2 (Required for requests to Chocolatey.org)
Getting latest version of the Chocolatey package for download.
Not using proxy.
Getting Chocolatey from https://community.chocolatey.org/api/v2/package/chocolatey/2.2.2.
Downloading https://community.chocolatey.org/api/v2/package/chocolatey/2.2.2 to C:\Users\HAMZA\AppData\Local\Temp\ch
colatey\chocoInstall\chocolatey.zip
Not using proxy.
Extracting C:\Users\HAMZA\AppData\Local\Temp\chocolatey\chocoInstall\chocolatey.zip to C:\Users\HAMZA\AppData\Local\T
emp\chocolatey\chocoInstall
Installing Chocolatey on the local machine
WARNING: Setting ChocolateyInstall Environment Variable on USER and not SYSTEM variables.
  This is due to either non-administrator install OR the process you are running is not being run as an Administrato
```



```
PS C:\Users\HAMZA> docker version
Client:
 Cloud integration: v1.0.35+desktop.5
 Version:           24.0.6
 API version:       1.43
 Go version:        go1.20.7
 Git commit:        ed223bc
 Built:             Mon Sep  4 12:32:48 2023
 OS/Arch:           windows/amd64
 Context:           default

Server: Docker Desktop 4.24.2 (124339)
 Engine:
  Version:          24.0.6
  API version:      1.43 (minimum version 1.12)
  Go version:       go1.20.7
  Git commit:       1a79695
  Built:            Mon Sep  4 12:32:16 2023
  OS/Arch:          linux/amd64
  Experimental:     false
 containerd:
  Version:          1.6.22
  GitCommit:        8165feabfdfe38c65b599c4993d227328c231fca
 runc:
  Version:          1.1.8
  GitCommit:        v1.1.8-0-g82f18fe
 docker-init:
  Version:          0.19.0
  GitCommit:        de40ad0
```

```
PS C:\Users\HAMZA\Desktop\Comptr rendu gummat avant le 05\Docker Part\web> docker -H tcp://your_windows_ip:2375 version
error during connect: Get "http://your_windows_ip:2375/v1.24/version": dial tcp: lookup your_windows_ip: no such host
Client:
 Cloud integration: v1.0.35+desktop.5
 Version:           24.0.6
 API version:       1.43
 Go version:        go1.20.7
 Git commit:        ed223bc
 Built:             Mon Sep  4 12:32:48 2023
 OS/Arch:           windows/amd64
 Context:           default
```

2. **Create and run a Docker image with an index.html web page**:

index.html

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Your Page Title</title>
</head>
<body>
    <header>
    <h1>Welcome BDCC</h1><span>Hamza</span>
    </header>

</body>
</html>
```

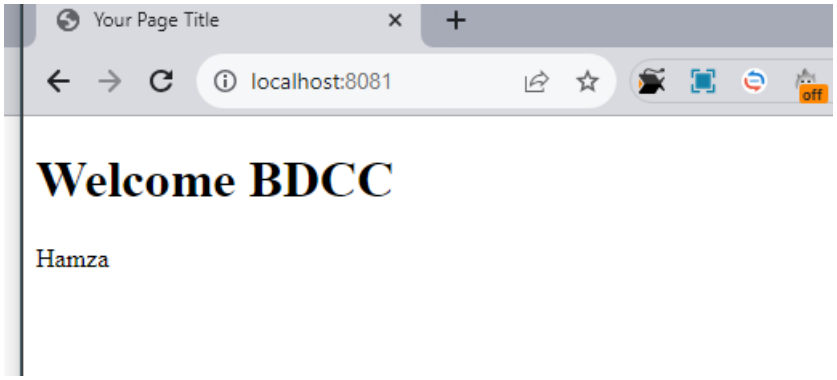
dockerfile

```
FROM nginx
COPY index.html /usr/share/nginx/html
EXPOSE 80
```

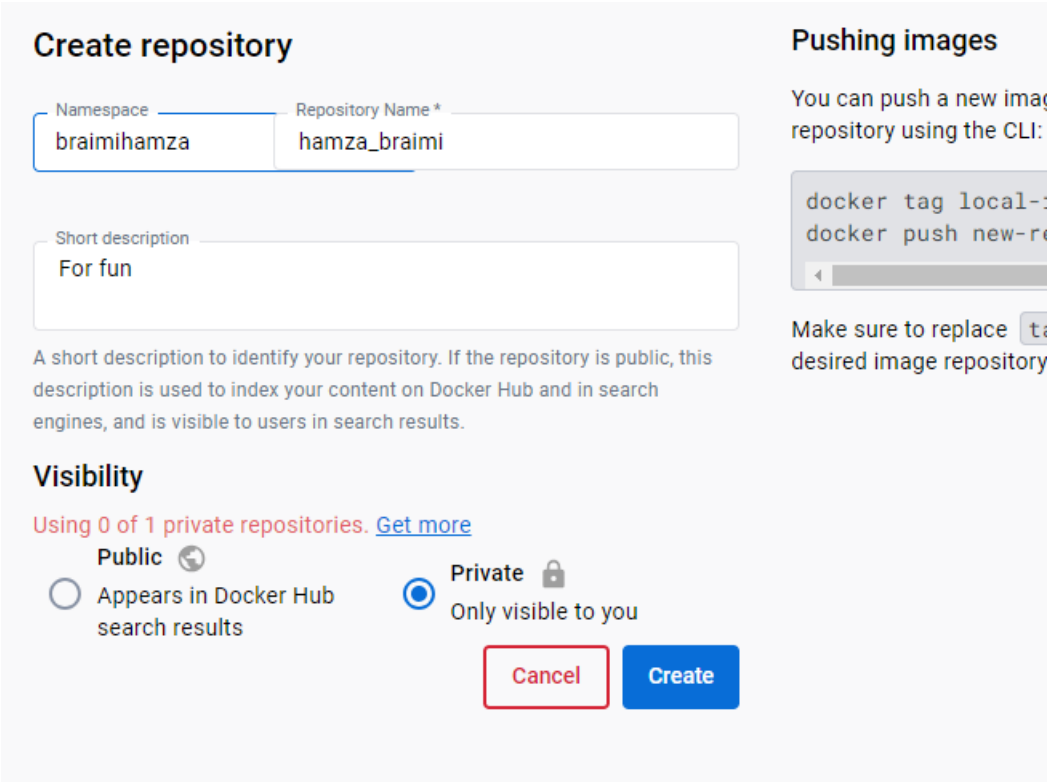
```
HAMZA@DESKTOP-RUTEDKE MINGW64 ~/Desktop/Comptr rendu gummat avant le 05/Docker P
art/web
$ docker build -t webapp2 .
#0 building with "default" instance using docker driver

#1 [internal] load .dockerignore
#1 transferring context: 2B done
#1 DONE 0.1s

#2 [internal] load build definition from dockerfile
#2 transferring dockerfile: 97B 0.0s done
#2 DONE 0.1s

#3 [internal] load metadata for docker.io/library/nginx:latest
#3 ...

#4 [auth] library/nginx:pull token for registry-1.docker.io
#4 DONE 0.0s

#3 [internal] load metadata for docker.io/library/nginx:latest
#3 DONE 3.1s

#5 [1/2] FROM docker.io/library/nginx@sha256:add4792d930c25dd2abf2ef9ea79de57809
7a1c175a16ab25814332fe33622de
#5 resolve docker.io/library/nginx@sha256:add4792d930c25dd2abf2ef9ea79de578097a1
c175a16ab25814332fe33622de
#5 ...

#6 [internal] load build context
#6 transferring context: 251B 0.0s done
#6 DONE 0.2s
```

```
HAMZA@DESKTOP-RUTEDKE MINGW64 ~/Desktop/Comptr rendu gummat avant le 05/Docker Part/web
$ docker images
REPOSITORY                TAG       IMAGE ID       CREATED          SIZE
webapp2                   latest    d6ae5be7613f   6 minutes ago    187MB
mysql                     latest    ae2502152260   2 months ago     574MB
phpmyadmin/phpmyadmin     latest    933569f3a9f6   3 months ago     562MB
docker/welcome-to-docker  latest    912b66cfd46e   4 months ago     13.4MB
hello-world               latest    9c7a54a9a43c   5 months ago     13.3kB
```

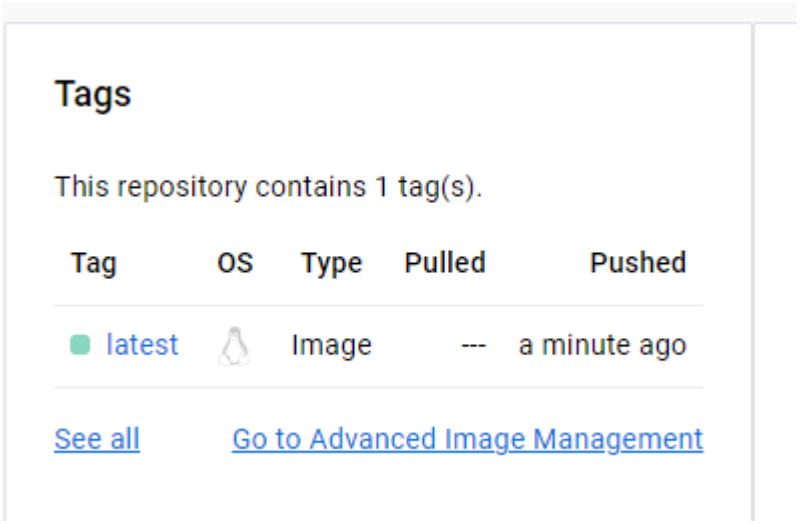
3. **Access to index.html using the host machine.**



4. **Launch image in private docker Hub**




```
HAMZA@DESKTOP-RUTEDKE MINGW64 ~/Desktop/Comptr rendu gummat avant le 05/Docker Part/web
$ docker login
Authenticating with existing credentials...
Login Succeeded

HAMZA@DESKTOP-RUTEDKE MINGW64 ~/Desktop/Comptr rendu gummat avant le 05/Docker Part/web
$ dokcer tage webapp2 braimihamza
/
hamza_braimi^C

HAMZA@DESKTOP-RUTEDKE MINGW64 ~/Desktop/Comptr rendu gummat avant le 05/Docker Part/web
$ docker tag webapp2 braimihamza/hamza_braimi

HAMZA@DESKTOP-RUTEDKE MINGW64 ~/Desktop/Comptr rendu gummat avant le 05/Docker Part/web
$ docker push braimihamza/hamza_braimi
Using default tag: latest
The push refers to repository [docker.io/braimihamza/hamza_braimi]
87d10c4a6e18: Preparing
97c766ccce63: Preparing
a7d081791a97: Preparing
9c7a17c7c18b: Preparing
572e6b1b9ddf: Preparing
f4e84f2b0154: Preparing
4ccbaf6751da: Preparing
f4e84f2b0154: Waiting
cb4596cc1454: Preparing
4ccbaf6751da: Waiting
cb4596cc1454: Waiting
9c7a17c7c18b: Mounted from library/nginx
572e6b1b9ddf: Mounted from library/nginx
a7d081791a97: Mounted from library/nginx
87d10c4a6e18: Pushed
f4e84f2b0154: Mounted from library/nginx
4ccbaf6751da: Mounted from library/nginx
cb4596cc1454: Mounted from library/nginx
97c766ccce63: Mounted from library/nginx
latest: digest: sha256:e487eca4b739d6489b26f6911f8f980efc7b9b9119e611e6cd35c5dcac46390a size: 1985

HAMZA@DESKTOP-RUTEDKE MINGW64 ~/Desktop/Comptr rendu gummat avant le 05/Docker Part/web
```



Conclusion

Docker is a powerful containerization platform that has revolutionized the way we develop and deploy applications. By packaging applications into lightweight containers, Docker makes it easier to achieve consistency, portability, and scalability.

In this practical guide, we have covered the essential steps to work with Docker, including installation, image management, and container orchestration. We have also explored Docker Hub, a repository for Docker images, and learned how to use it to find and share images.