TP Docker MySQL

Nom: KADDOUR BAKIR Riham

Objectif

Mettre en place un conteneur MySQL avec Docker, créer une base de données, insérer des données et tester la persistance à travers le redémarrage du conteneur.

Étapes et commandes

0.1 Créer un conteneur MySQL propre

```
docker run -d
--name mysql-demo
-e MYSQL_ROOT_PASSWORD=pass123
-v mysql_data:/var/lib/mysql
mysql:8
```

```
kaddourbakirriham@cloudshell:~/2decembre (discovergirl)$ docker run -d \
  --name mysql-demo \
  -e MYSQL_ROOT_PASSWORD=pass123 \
  -v mysql_data:/var/lib/mysql \
  mysql:8
2f90072a021893b56b2a65d4bd1f71c99b692031d0f08807e0aa99d411b513e8
```

Vérifier que le conteneur fonctionne

```
docker ps
```

```
kaddourbakirriham@cloudshell:~/2decembre (discovergirl)$ docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED          STATUS
       PORTS                                  NAMES
2f90072a0218   mysql:8                              "docker-entrypoint.s…"   25 seconds ago   Up 25 sec
onds   3306/tcp, 33060/tcp                    mysql-demo
b4b9bbefcf88   gcr.io/k8s-minikube/kicbase:v0.0.48   "/usr/local/bin/entr…"   34 minutes ago   Up 34 mir
utes   127.0.0.1:32768->22/tcp, 127.0.0.1:32769->2376/tcp, 127.0.0.1:32770->5000/tcp, 127.0.0.1:32771->8
443/tcp, 127.0.0.1:32772->32443/tcp   minikube
```

Se connecter au conteneur MySQL

```
docker exec -it mysql-demo mysql -uroot -p
```

```
kaddourbakirriham@cloudshell:~/2decembre (discovergirl)$ docker exec -it mysql-demo mysql -uroot -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.4.7 MySQL Community Server - GPL

Copyright (c) 2000, 2025, 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> CREATE DATABASE testdb;
```

Créer une base de données

```
CREATE DATABASE testdb;
SHOW DATABASES;
```

```
mysql> CREATE DATABASE testdb;
Query OK, 1 row affected (0.01 sec)

mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
| testdb             |
+--------------------+
5 rows in set (0.01 sec)
```

Créer une table et insérer des données

```
USE testdb;
CREATE TABLE users (
id INT AUTO_INCREMENT PRIMARY KEY,
name VARCHAR(50),
email VARCHAR(50)
);
INSERT INTO users (name, email) VALUES
('Alice', '[alice@example.com](mailto:alice@example.com)'),
('Bob', '[bob@example.com](mailto:bob@example.com)');
SELECT * FROM users;
```

```
mysql> USE testdb;
INCREMENT PRIMARY KEY,
    name VARCHAR(50),
    eDatabase changed
mysql>
mysql> CREATE TABLE users (
    ->     id INT AUTO_INCREMENT PRIMARY KEY,
    ->     name VARCHAR(50),
    ->     email VARCHAR(50)
    -> );
l) VALUES
('Alice', 'alice@example.com'),
('Bob', 'bob@example.com');

SELECT * FROM users;
Query OK, 0 rows affected (0.02 sec)
```

Arrêter et redémarrer le conteneur

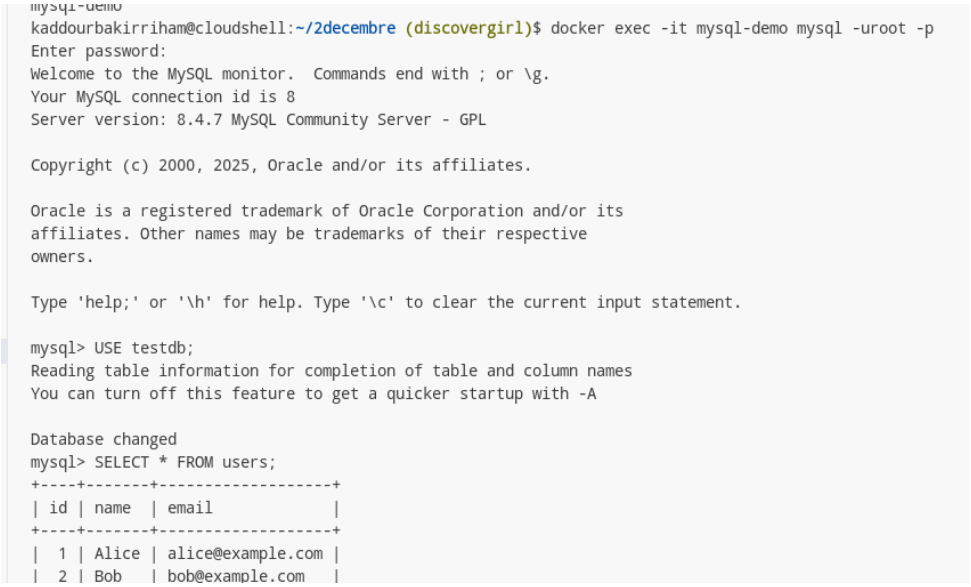
```
docker stop mysql-demo
docker start mysql-demo
```

```
kaddourbakirriham@cloudshell:~/2decembre (discovergirl)$ docker stop mysql-demo
mysql-demo
kaddourbakirriham@cloudshell:~/2decembre (discovergirl)$ docker start mysql-demo
mysql-demo
```

Vérifier la persistance des données

```
docker exec -it mysql-demo mysql -uroot -p
USE testdb;
SELECT * FROM users;
```

Toutes les données doivent toujours être présentes.



Créer un deuxième conteneur en partageant les données

```
docker stop mysql-demo
docker run -d
--name mysql-demo2
-e MYSQL_ROOT_PASSWORD=pass123
-v mysql_data:/var/lib/mysql
mysql:8
```

Accéder aux données depuis le deuxième conteneur

```
docker exec -it mysql-demo2 mysql -uroot -p
USE testdb;
SELECT * FROM users;
```

Les mêmes données doivent être accessibles.

```
kaddourbakirriham@cloudshell:~/2decembre (discovergirl)$ docker exec -it mysql-demo2 mysql -uroot -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.4.7 MySQL Community Server - GPL

Copyright (c) 2000, 2025, 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> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
| testdb             |
+--------------------+
5 rows in set (0.01 sec)

mysql> USE testdb;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> SELECT * FROM users;
+----+-------+-------------------+
| id | name  | email             |
+----+-------+-------------------+
|  1 | Alice | alice@example.com |
|  2 | Bob   | bob@example.com   |
```

0.2 Page statique HTML avec Nginx

Objectif : Créer une page HTML statique et la servir avec Nginx dans un conteneur Docker.

1. Lancer un conteneur Nginx avec le port 8080 exposé :

   ```
   docker run -d -p 8080:80 --name site nginx
   ```

2. Accéder au conteneur :

```
docker exec -it site bash
```

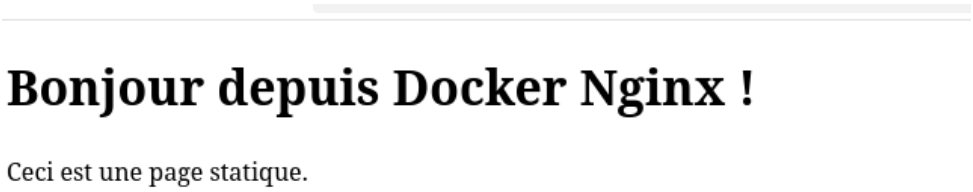
3. Se placer dans le répertoire où Nginx sert les fichiers :

```
cd /usr/share/nginx/html
```

4. Créer le fichier `index.html` :

```
echo '<!DOCTYPE html>
<html>
<head>
    <title>Ma page statique</title>
</head>
<body>
    <h1>Bonjour depuis Docker Nginx !</h1>
    <p>Ceci est une page statique.</p>
</body>
</html>' > index.html
```

5. Vérifier l'accès à la page via l'URL Web Preview :



0.3 2. Application Java multi-étapes Docker

Objectif : Compiler et exécuter un programme Java dans un conteneur Docker.

1. Dockerfile utilisé :

```
# Étape 1 : compilation
FROM eclipse-temurin:21-jdk AS build
WORKDIR /app
COPY Hello.java .
RUN javac Hello.java

# Étape 2 : exécution
FROM eclipse-temurin:21-jre
WORKDIR /app
COPY --from=build /app/Hello.class .
CMD ["java", "Hello"]
```

2. Construire l'image :

```
docker build -t hello-java .
```

3. Lancer le conteneur :

```
docker run  hello-java
```