

LabXpert docker part

Conteneur (Container) :

Un conteneur est une instance isolée et exécutable d'une application, avec tout ce dont elle a besoin pour fonctionner, y compris le code, les bibliothèques et les dépendances. Les conteneurs sont légers, portables et peuvent fonctionner de manière cohérente dans différents environnements.

Dockerfile :

Un Dockerfile est un fichier texte qui contient les instructions pour créer une image Docker. Il spécifie les étapes nécessaires pour configurer l'environnement dans lequel votre application s'exécutera. Les Dockerfiles sont utilisés avec la commande `docker build` pour créer des images Docker.

Réseau (Network) :

Un réseau Docker est un ensemble de conteneurs Docker qui peuvent communiquer entre eux, soit sur la même machine hôte ou sur des machines hôtes différentes. Les réseaux Docker permettent aux conteneurs de se connecter les uns aux autres et de communiquer, ce qui est essentiel pour les applications distribuées.

Image Docker (Docker Image) :

Une image Docker est un package léger et autonome qui contient tout le nécessaire pour exécuter une application : le code, les bibliothèques, les dépendances, les variables d'environnement et les fichiers de configuration. Les images Docker sont créées à partir d'un Dockerfile et peuvent être partagées et utilisées pour déployer rapidement des applications.

Volume Docker (Docker Volume) :

Un volume Docker est un mécanisme pour persister les données générées et utilisées par les conteneurs Docker. Les volumes Docker sont montés sur des emplacements spécifiques dans le système de fichiers du conteneur, ce qui permet aux données d'être stockées de manière persistante même lorsque le conteneur est arrêté ou supprimé. Les volumes Docker sont utiles pour le stockage des données des bases de données, des fichiers de logs, des fichiers de configuration, etc.

Dans cette brief j'ai fait les commandes suivants pour créer et relier les conteneur:

Cette commande créer le container avec le nom `postgres-container` qui a pris les informations de postgres (password, username et db_name) et il a pris le port externe 5432 et le volume anonyme et téléchargez l'image postgres .

```
(maro@maro) - [~/IdeaProjects/labxpert part2]
$ docker run --name postgres-container -e POSTGRES_PASSWORD=1111 -e POSTGRES_USERNAME=postgres -e POSTGRES_DB=test_docker -d -p 5432:5432 -v /var/lib/postgresql/data postgres
```

Cette commande et builder le jar file de l'application:

```
(maro@Maro)-[~/IdeaProjects/labxpert part2]  
$ ./mvnw clean package -DskipTests
```

Cette screen représente le dockerfile:

Il a pris le jdk 8 et le jar qui déjà builder dans le fichier target et le port 8081 et la commande d'exécution

```
Dockerfile x  
1  FROM openjdk:8-jdk  
2  
3  WORKDIR /app  
4  
5  COPY target/labxpert-0.0.1-SNAPSHOT.jar /app/labxpert.jar  
6  
7  EXPOSE 8081  
8  
9  CMD ["java", "-jar", "labxpert.jar"]
```

Cette commande et builder images d'application:

```
(maro@Maro)-[~/IdeaProjects/labxpert part2]  
$ docker build -t labxpert-app .  
□
```

Cette commande créer un network:

```
(maro@Maro)-[~/IdeaProjects/labxpert part2]  
$ docker network create labxpert-net
```

Cette commande connect le network avec container de postgres:

```
(maro@Maro)-[~/IdeaProjects/labxpert part2]  
$ docker network connect labxpert-app postgres-lab-xpert
```

Cette commande run l'image d'application relier avec le network qui déjà connecté avec le container postgres:

```
(maro@Maro)-[~/IdeaProjects/labxpert part2]  
$ docker run -p 8088:8081 --net labxpert-net labxpert-app
```

Il doit changer le domaine URL postgres par la nom de container postgres:

```
ing.jpa.properties.hibernate.dialect=org.hibernate.dialect.PostgreSQL81Dialect
ing.datasource.url=jdbc:postgresql://postgres-labxpert:5432/db-lab-xpert
```

Pour push un image dans docker hub:

Il doit créer un repository tout d'abord et après créer un étiquette pour l'image:

```
(maro@Maro) - [~/IdeaProjects/labxpert part2]
$ docker tag labxpert-app marouane01/labxpert-docker-part:latest
```

Il pousse dans le repository créer:

```
(maro@Maro) - [~/IdeaProjects/labxpert part2]
$ docker push marouane01/labxpert-docker-part:latest
```