SIMPLON.CO

Jobintech

Documentation technique du projet LabXpert - Déployer l'API Rest sur Docker

Réalisé par

TKARKIB Lhassan

2023-2024

TABLE DES MATIÈRES

Documentation 2
1 Introduction 2
2 Etude technique et technologique 2
2.1 Docker 2
2.2 Image Docker 2
2.3 Docker Hub 3
3 Configuration 4
4 Conclusion 6

1 Introduction

Dans le paysage actuel du développement logiciel, le déploiement d'applications est souvent une tâche complexe et difficile. Les méthodes de déploiement traditionnelles peuvent entraîner des problèmes tels que des conflits de dépendances, des incohérences d'environnement et des échecs de déploiement. Pour relever ces défis, la conteneurisation a émergé comme une solution puissante, avec Docker en tête en tant que norme de facto pour la conteneurisation.

Ce projet se concentre sur la dockerisation d'une application backend Spring Boot REST API à l'aide de Docker et Dockerfile. Docker fournit une solution de conteneurisation indépendante de la plate-forme qui permet aux développeurs de packager des applications et leurs dépendances dans des conteneurs légers et portables. En encapsulant les applications dans des conteneurs, Docker permet une cohérence dans le déploiement à travers différents environnements, du développement à la production.

2 Etude technique et technologique

2.1 Docker

FIGURE 1 – Docker Logo

Docker est une plateforme de conteneurisation qui permet aux développeurs de créer, de distribuer et d'exécuter des applications dans des conteneurs. Ces conteneurs offrent un environnement isolé pour exécuter des applications, ce qui garantit une portabilité et une fiabilité accrues. Les principaux avantages de Docker sont :

- **Isolation** : Les conteneurs Docker fournissent une isolation des processus, des systèmes de fichiers et des réseaux, ce qui permet d'éviter les conflits entre les différentes applications et les dépendances.
- **Portabilité** : Les conteneurs Docker sont légers et portables, ce qui signifie qu'ils peuvent être exécutés de manière cohérente sur n'importe quel environnement compatible avec Docker, que ce soit un ordinateur portable, un serveur local ou le cloud.
- **Productivité** : Docker simplifie le processus de développement, de test et de déploiement des applications en fournissant des environnements reproductibles et des workflows efficaces basés sur des conteneurs.

2.2 Image Docker

Une image Docker est un package léger et autonome qui contient tout le nécessaire pour exécuter une application : le code source, les dépendances, les bibliothèques et les paramètres d'exécution. Les images Docker sont utilisées pour créer des conteneurs Docker en spécifiant

FIGURE 2 – Docker Logo

les commandes d'exécution et les configurations nécessaires. Les principales caractéristiques des images Docker sont :

- **Reproductibilité** : Les images Docker garantissent la reproductibilité des environnements d'exécution, ce qui facilite le développement et le déploiement des applications dans des environnements divers.
- **Modularité** : Les images Docker sont construites selon le principe de modularité, ce qui signifie qu'elles peuvent être composées de manière flexible à partir d'autres images de base ou de composants spécifiques.
- **Gestion des Versions** : Les images Docker peuvent être versionnées et gérées à l'aide de registres Docker, ce qui permet de suivre les changements, de revenir à des versions antérieures et de partager les images avec d'autres utilisateurs.

2.3 Docker Hub

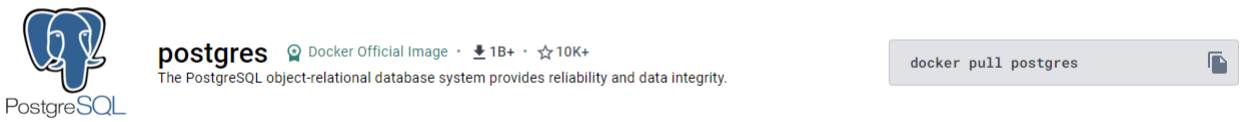
FIGURE 3 – Docker Logo

Docker Hub est un service cloud fourni par Docker qui agit comme un registre de conteneurs Docker. Il permet aux développeurs de stocker, de partager et de gérer des images Docker, ainsi que de collaborer avec d'autres membres de la communauté Docker. Les principales fonctionnalités de Docker Hub sont :

- **Stockage d'Images** : Docker Hub permet aux développeurs de stocker des images Docker publiques et privées dans des dépôts sur le cloud.
- **Partage d'Images** : Les développeurs peuvent partager leurs images Docker avec d'autres membres de la communauté Docker en les publiant sur Docker Hub.
- **Intégration Continue/Déploiement Continu (CI/CD)** : Docker Hub prend en charge l'intégration continue et le déploiement continu en permettant l'automatisation des tests, de la construction et du déploiement d'images Docker à partir de référentiels de code source.

3 Configuration

***l'importation de l'image de postgres du dockerhub :**



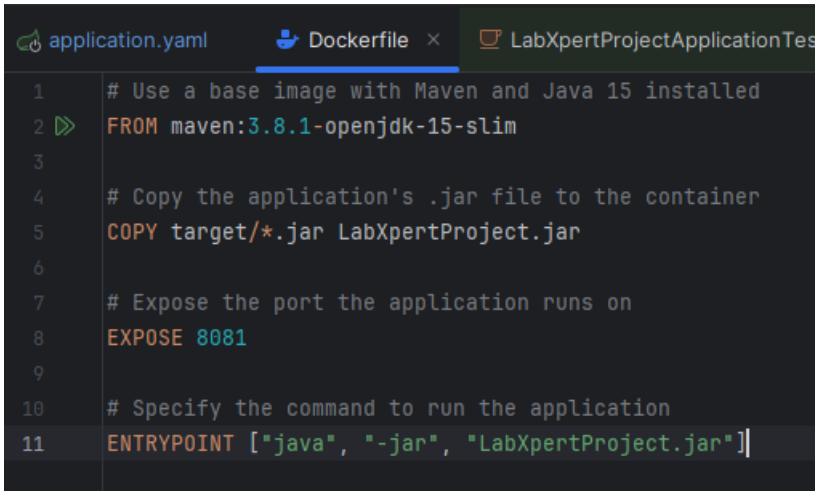
***Creation de reseau :**

```
PS C:\Users\Dell Latitude 5440\Desktop\java+projects\LabXpertSecurity> docker network create labxpert_network
```

*** execution de l'image de postgres :**

```
PS C:\Users\Dell Latitude 5440\Desktop\java+projects\LabXpertSecurity> docker run --network=labxpert_network -d -p 4444:5432 -e POSTGRES_PASSWORD=50145014 -e POSTGRES_USER=postgre
r_data:/var/lib/postgresql/data --name=postgres_con postgres
```

*** Docker file :**



*** Creation de l'image de l'application :**

```
PS C:\Users\Dell Latitude 5440\Desktop\java+projects\LabXpertSecurity> docker build -t labxpertproject_img
```

*** Execution de l'image de l'application :**

```
PS C:\Users\Dell Latitude 5440\Desktop\java+projects\LabXpertSecurity> docker run --network=labxpert_network -d -p 8082:8081 --name=labxpertproject_con labxpertproject_img
8438eb562e3f168e6af49a756c8b067339c84eafe53f92fa7f36030a6ce80c4c
```

```
PS C:\Users\Dell Latitude 5440\Desktop\java+projects\LabXpertSecurity>  docker tag labxpertproject_img ryukiro/labxpert:labxpertapp
PS C:\Users\Dell Latitude 5440\Desktop\java+projects\LabXpertSecurity> docker push ryukiro/labxpert:labxpertapp
The push refers to repository [docker.io/ryukiro/labxpert]
5ef2b61227a1: Pushed
490fd7c1d2c6: Pushed
e8f9a2b449df: Pushed
e8055628a9a8: Pushed
2e1389b4933a: Pushed
afe38b9c520e: Pushed
2651e73b1464: Pushed
7e718b9c0c8c: Pushed
labxpertapp: digest: sha256:161512abd1ab087108b144b783b56a3c97cbe515d9a4fadd44a370625be92a8e size: 2208
```

*** Push l'images de l'application dans docker hub ropsitorie :**

*** Push l'images de base de donnees dans docker hub ropsitorie :**

```
PS C:\Users\Dell Latitude 5440\Desktop\java+projects\LabXpertSecurity> docker tag postgres ryukiro/labxpert:labxpertpostgres
PS C:\Users\Dell Latitude 5440\Desktop\java+projects\LabXpertSecurity> docker push ryukiro/labxpert:labxpertpostgres
The push refers to repository [docker.io/ryukiro/labxpert]
027fd566b2ac: Layer already exists
50fdff915195: Layer already exists
e74743b5b995: Layer already exists
39c411664fe7: Layer already exists
de04134f2aa6: Layer already exists
151fdb0659e6: Layer already exists
66535b2ade3c: Layer already exists
b11b371ada64: Layer already exists
ec1f0e5ec5c7: Layer already exists
fee286e0678a: Layer already exists
4e75377ece4e: Layer already exists
2a941506b129: Layer already exists
3e22e5fb0c85: Layer already exists
ceb365432eec: Layer already exists
labxpertpostgres: digest: sha256:337d71ccfe33b75ec608302dc630ce035918c5cd88c4b5a7234a00128e183a35 size: 3247
```

4 Conclusion

En conclusion, la dockerisation de notre application Spring Boot avec PostgreSQL représente une avancée significative dans le déploiement et la gestion de notre système. Grâce à Docker, nous avons pu encapsuler notre application et sa base de données dans des conteneurs légers et portables, offrant ainsi une solution de déploiement cohérente et efficace.

L'utilisation de Docker nous a permis de résoudre plusieurs défis de déploiement, notamment la gestion des dépendances, la cohérence de l'environnement et la portabilité entre les différentes phases de développement. De plus, en utilisant Docker Hub comme registre d'images, nous avons facilité le partage et la distribution de nos images Docker avec l'équipe de développement et les parties prenantes.

En adoptant la containerisation avec Docker, notre équipe a gagné en agilité et en efficacité dans le déploiement de notre application. Nous sommes désormais mieux équipés pour faire face aux défis futurs du déploiement logiciel et pour assurer la stabilité et la performance de notre système dans des environnements variés.

En définitive, ce projet nous a permis de mieux comprendre les avantages de Docker dans le développement et le déploiement d'applications, et nous sommes convaincus que cette technologie continuera à jouer un rôle essentiel dans notre stratégie de déploiement à l'avenir.