

1 Dockerisation de l'Application Backend Spring Boot REST API

1. Introduction

Ce projet avait pour objectif de comprendre et d'appliquer les concepts de base de Docker pour le déploiement d'applications. Nous avons également cherché à développer nos compétences dans la gestion des conteneurs.

2. Configuration de Docker

La première étape a été d'installer Docker sur nos machines pour pouvoir commencer le processus de dockerisation de notre application.

+ Creation de reseau

Un réseau Docker nommé "labxpert_network" est créé pour connecter les conteneurs de l'application Spring Boot et de la base de données PostgreSQL.

```
C:\workplace\JobIntech\sécurité\LabXpertV2>docker network create labxpert_network
e4b2efa6441a6b20477694a280f3a7f4a2f569e078efc2e37e68b2c93025bc27
```

+ Creation image de base de donnee Postgres

la commande **docker pull postgres** pour télécharger l'image PostgreSQL à partir de Docker Hub.

+ Creation de conteneur de base de donnee Postgres

- Un conteneur PostgreSQL est créé avec le nom "LabXpert1 " et connecté au réseau "labxpert_network".
- Il expose le port 4444 localement et définit un mot de passe pour l'accès à la base de données.

```
C:\workplace\JobIntech\sécurité\LabXpertV2>docker run --network=labxpert_network -d -p 4444:5432 -e POSTGRES_PASSWORD=123456 -e POSTGRES_USER=postgres -e POSTGRES_DB=LabXpert1 -v C:\postgres_container_data:/var/lib/postgresql/data --name=postgres_con postgres
9934c7355e619b1ac64570110ed3dd7a489c191fd34e3dccc9c8eaac4d98cf8c
```

+ Dockerfile

Dockerfile est un fichier utilisé pour définir l'environnement d'exécution de notre application dans un conteneur Docker.

- Le Dockerfile spécifie l'utilisation de l'image OpenJDK version 15.

- Il copie le fichier JAR de l'application dans le conteneur.
- L'entrée principale du conteneur est configurée pour exécuter l'application Spring Boot.

The screenshot shows an IDE with a project named 'LabXpertV2'. The left sidebar displays a file tree with folders 'repository', 'service', and 'target'. The 'target' folder is expanded, showing files like '.gitignore', 'DocHALIMAELAMRI.pdf', 'Dockerfile', 'Jenkinsfile', 'LabXpertProject.iml', 'mvnw', 'mvnw.cmd', and 'pom.xml'. The 'Dockerfile' is selected and its content is shown in the main editor. The Dockerfile content is as follows:

```

1 #Use a base image with Maven and Java 15 installed
2 FROM maven:3.8.1-openjdk-15-slim
3
4 #Copy the application's .jar file to the container
5 COPY target/*.jar LabXpertProject.jar
6
7 #Expose the port the application runs on
8 EXPOSE 8081
9
10 #Specify the command to run the application
11 ENTRYPOINT ["java", "-jar", "LabXpertProject.jar"]

```

✚ Construction de l'Image Docker

The screenshot shows a terminal window with the command prompt 'C:\workplace\JobIntech\sécurité\LabXpertV2> docker build -t labxpertproject_img .' and the output of the build process. The output shows the progress of the build, including the transfer of the Dockerfile, the resolution of the base image, and the final build completion.

```

C:\workplace\JobIntech\sécurité\LabXpertV2> docker build -t labxpertproject_img .
[+] Building 240.9s (8/8) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 372B
=> [internal] load metadata for docker.io/library/maven:3.8.1-openjdk-15-slim
=> [auth] library/maven:pull token for registry-1.docker.io
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [internal] load build context
=> => transferring context: 66.40MB
=> [1/2] FROM docker.io/library/maven:3.8.1-openjdk-15-slim@sha256:133a24a6fb813c7420072ae2ece015a0a5d62835b94e5143b11a92957edac0d7
=> => resolve docker.io/library/maven:3.8.1-openjdk-15-slim@sha256:133a24a6fb813c7420072ae2ece015a0a5d62835b94e5143b11a92957edac0d7
=> => sha256:6df176b870d2923915a706c7851fa5806de8570ff3c5dad8029e518e98d7c89a 2.00kB / 2.00kB
=> => sha256:589ba3001ded2440a1e16209e9ae141f2043bf6f28715a9d70dc09b9c332931d 8.32kB / 8.32kB
=> => sha256:133a24a6fb813c7420072ae2ece015a0a5d62835b94e5143b11a92957edac0d7 549B / 549B

```

✚ Exécution de L'image

```
C:\workplace\JobIntech\sécurité\LabXpertV2>docker run --network=labxpert_network -d -p 8082:8081 --name=labxpertproject_con labxpertproject_img_4a7cba532f0dc580ecf7e3302509ae848548749e56acf4d3cf3df7eee42d8960
```

🚀 Push image to Docker Hub

Creation de tag pour database image et Push l'image a docker hub

```
C:\workplace\JobIntech\sécurité\LabXpertV2>docker tag postgres halima0/labxpert:labxpertpostgres

C:\workplace\JobIntech\sécurité\LabXpertV2>docker push halima0/labxpert:labxpertpostgres
The push refers to repository [docker.io/halima0/labxpert]
027fd566b2ac: Mounted from library/postgres
50fdff915195: Mounted from library/postgres
e74743b5b995: Mounted from library/postgres
39c411664fe7: Mounted from library/postgres
de04134f2aa6: Mounted from library/postgres
151fdb0659e6: Mounted from library/postgres
66535b2ade3c: Mounted from library/postgres
b11b371ada64: Mounted from library/postgres
ec1f0e5ec5c7: Mounted from library/postgres
```

Creation de tag pour notre application image et Push l'image a docker hub

```
C:\workplace\JobIntech\sécurité\LabXpertV2>docker tag labxpertproject_img halima0/labxpert:labxpertapp

C:\workplace\JobIntech\sécurité\LabXpertV2>docker push halima0/labxpert:labxpertapp
The push refers to repository [docker.io/halima0/labxpert]
1366aa408142: Pushed
490fd7c1d2c6: Pushed
e8f9a2b449df: Pushed
e8055628a9a8: Pushed
2e1389b4933a: Pushed
29d4ff6631bb: Pushed
afe38b9c520e: Pushed
2651e73b1464: Pushed
7e718b9c0c8c: Pushed
labxpertapp: digest: sha256:2c385192aca43d57758b7c1e718e00140861f9adb6251a6d6e61d2629ba19b5a size: 2208

C:\workplace\JobIntech\sécurité\LabXpertV2>
```

Conclusion

Ce projet nous a permis de mieux comprendre l'importance et les avantages de l'utilisation de Docker pour le déploiement d'applications. Nous avons acquis des compétences précieuses dans la gestion des conteneurs et dans la configuration d'environnements d'exécution. La dockerisation de notre application backend Spring Boot REST API représente une étape importante dans notre parcours de développement logiciel.