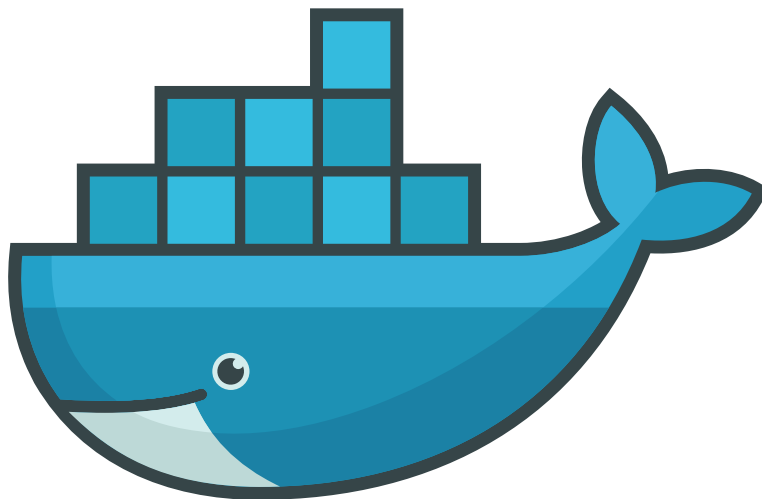


Brief 11: LabXpert - Déployer l'API Rest sur Docker

Docker:



Docker est une plateforme logicielle permettant de créer, déployer et gérer des applications dans des conteneurs. Les conteneurs sont des environnements isolés qui contiennent tout ce dont une application a besoin pour fonctionner, comme le code, les bibliothèques et les dépendances. Docker facilite le déploiement d'applications car les conteneurs sont portables et peuvent fonctionner de manière cohérente sur n'importe quel système compatible avec Docker.

L'application LabXpert et Docker:

Le fichier **Dockerfile** :

Un *Dockerfile* est un fichier texte qui contient les instructions pour construire une image Docker. Il définit un environnement de conteneur qui peut être utilisé pour créer des conteneurs qui exécutent votre application.

Voici quelques utilisations courantes d'un Dockerfile :

- Définir l'environnement de l'application : Vous pouvez spécifier l'image de base à utiliser pour votre application, par exemple une image spécifique de Java ou de Node.js.
- Installer les dépendances nécessaires : Vous pouvez utiliser les instructions du Dockerfile pour installer toutes les dépendances requises par votre application.
- Copier le code source de l'application dans l'image : Vous pouvez copier votre code source dans l'image Docker, ce qui permet à l'application de s'exécuter dans le conteneur.
- Configurer l'application : Vous pouvez définir des variables d'environnement, exposer des ports, définir une commande par défaut à exécuter lorsque le conteneur démarre, etc.
- Automatiser le processus de construction : Avec un Dockerfile, vous pouvez automatiser le processus de construction de votre image Docker, ce qui facilite le déploiement et la distribution de votre application.

Dockerfile dans **LabXpert** :

FROM openjdk:17-jdk

WORKDIR /app

COPY target/labxpert-0.0.1-SNAPSHOT.jar /app/labxpert-0.0.1-SNAPSHOT.jar

ENTRYPOINT ["java", "-jar", "labxpert-0.0.1-SNAPSHOT.jar"]

- **FROM** openjdk:17-jdk : Cette instruction définit l'image de base à utiliser pour le conteneur. Ici, vous utilisez l'image openjdk avec la version 17-jdk. Cette image contient l'environnement Java Development Kit (JDK) version 17, qui est nécessaire pour exécuter votre application Java.
- **WORKDIR** /app : Cette instruction définit le répertoire de travail dans le conteneur. Toutes les instructions suivantes qui font référence à un chemin d'accès relatif seront exécutées par rapport à ce répertoire. Ici, vous définissez le répertoire de travail comme /app.
- **COPY** target/labxpert-0.0.1-SNAPSHOT.jar /app/labxpert-0.0.1-SNAPSHOT.jar : Cette instruction copie le fichier labxpert-0.0.1-SNAPSHOT.jar de votre répertoire target local dans le répertoire /app du conteneur. Ce fichier JAR est l'artefact de votre application, qui peut être exécuté pour démarrer votre application.
- **ENTRYPOINT** ["java", "-jar", "labxpert-0.0.1-SNAPSHOT.jar"] : Cette instruction définit la commande par défaut à exécuter lorsque le conteneur démarre. Ici, vous exécutez la commande java -jar labxpert-0.0.1-SNAPSHOT.jar pour démarrer votre application.

Les commande pour cree une image docker mais apre le pull de l'image postgres :

- cmd1 : pour extraire une image déjà existante telle que Postgres.

docker pull postgres

- cmd2 : create network.

docker network create new-network-yassin

- cmd3: pou cree un contenaire Postgres dans le network

docker run --name postgres -bases --network new-network-yassin -e POSTGRES_PASSWORD=yasso DB=db-lab-xpert -e POSTGRES_USER=postgres -p 4444:5432 --restart=on-failure -d postgres

- cmd4 : image pour le project labxpert

docker build -t myapp .

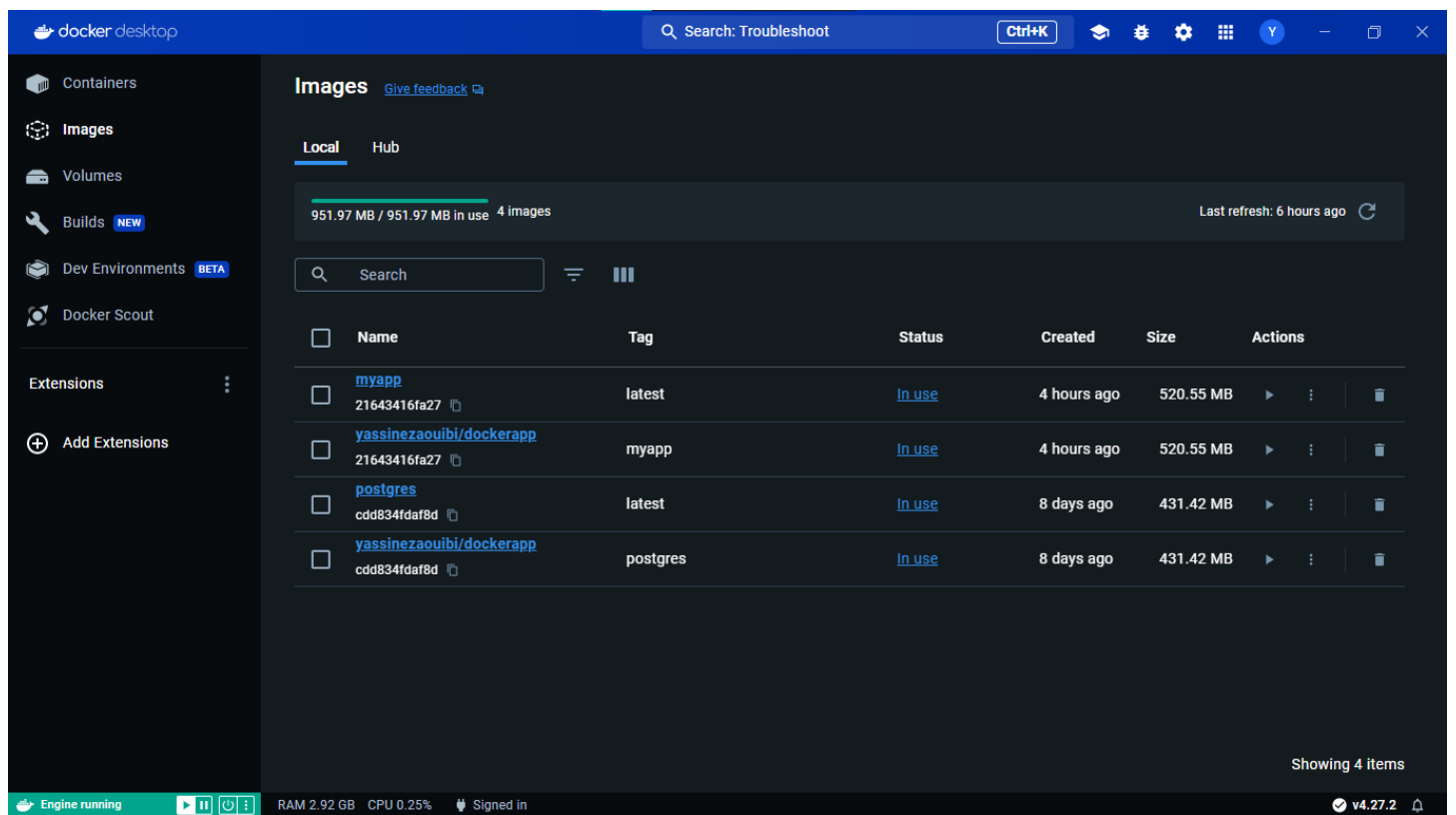
- cmd4: create network

docker network create new-network-yassin

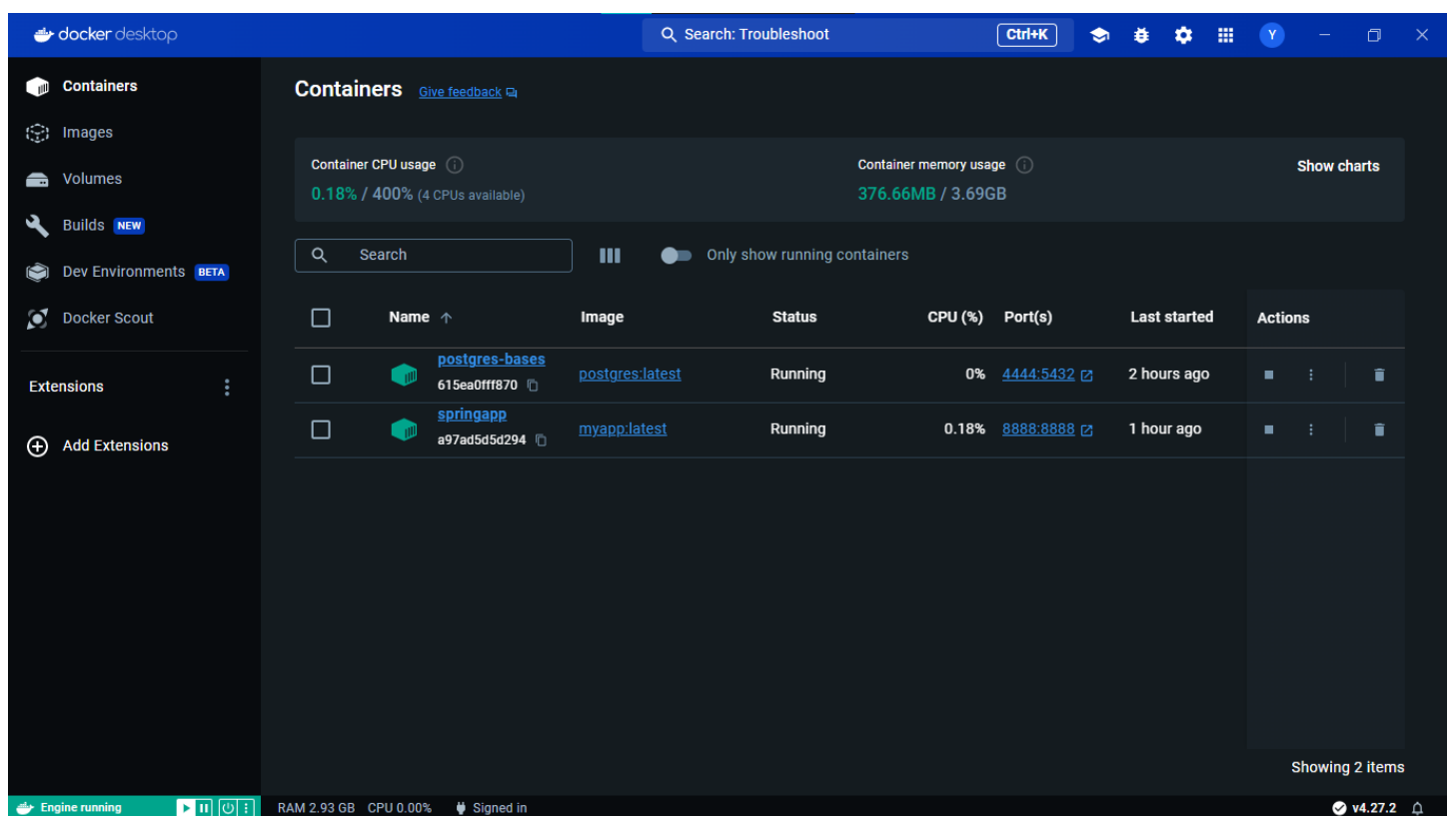
- cmd5 : créer et exécuter un nouveau conteneur Docker basé

docker run -d -p 8888:8888 --name=myapp --network new-network-yassin myapp:tag

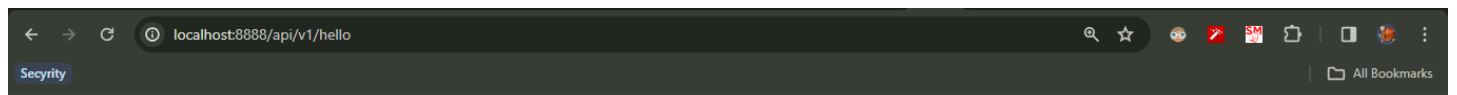
Les images Docker :



Les Conteneurs Docker :



L'Application dans le serveur :



Hello Yassin, you have to see this text in the
docker running side
