Lancez un conteneur PHPMyAdmin en connectant le même réseau Docker :

docker run -d --name phpmyadmin --network gestion-stock-network \

-e PMA\_HOST=mysql-container \

-p 8081:80 phpmyadmin

Lancer conteneur mysql

docker run --name mysql-container --network gestion-stock-network -e MYSQL\_ROOT\_PASSWORD=admin\_password -e MYSQL\_DATABASE=gestion\_stock -p 3306:3306 -d mysql:8.0

inspecter le réseau

docker network inspect gestion-stock-network

se connecter à la base de donnée

$ docker exec -it mysql-container mysql -u root -p

Enter password: admin\_password

mysql>

**Construire l'image Docker :**

mvn clean package

docker build -t gestion-stock-backend .

**Lancer le conteneur Spring Boot : dans le même network**

docker run -d --name springboot-backend --network gestion-stock-network \

-p 8080:8080 gestion-stock-backend

**Supprimez le conteneur existant avec le nom springboot-backend :**

**docker rm springboot-backend**

Redémarrez ensuite le conteneur avec la commande précédente :

docker run -d --name springboot-backend --network gestion-stock-network -p 8080:8080 gestion-stock-backend

Construire l’image du front angular

docker build -t angular-frontend .

1. Lancer le conteneur

docker run -d --name angular-frontend --network gestion-stock-network -p 4200:4200 angular-frontend

backend nodejs

consstruire l’image

docker build -t product-management-backend .

lancer le conteneur

docker run -d -p 5000:5000 --name product-management-backend product-management-backend

pour supprimer si y’a une erreur docker rm -f product-management-backend

et relancer docker run -d -p 5000:5000 --name product-management-backend product-management-backend

pour le front de react

construire l’image

docker build -t product-management-frontend .

lancer le conteneur

docker run -d -p 3000:80 --name product-management-frontend product-management-frontend

30/12/2024

Docker 100%

Tout fonctionne bien

Nb: pour lancer mvn clean install à partir intellij ( il faut d’abord lancer Xamp MySQL)

Ensuite lancer la commander docker-compose up –build ( qui va lancer le back , front et mysql )

Docker

Pour arrêter tous les conteneurs en cours d'exécution, vous pouvez utiliser la commande suivante :

docker-compose down

Cette commande arrête et supprime tous les conteneurs, réseaux et volumes associés à votre projet. Si vous ne voulez pas supprimer les volumes (comme les données

MySQL), vous pouvez ajouter l'option --volumes : ( ne pas pour ne pas perdre les données )

docker-compose down –volumes

**Redémarrer les conteneurs Docker**

Pour relancer les conteneurs après les avoir arrêtés, vous pouvez utiliser la commande suivante :

docker-compose up

Cela va lancer vos conteneurs tels qu'ils étaient définis dans votre fichier docker-compose.yml.

**Redémarrer avec construction (si nécessaire)**

Si vous avez apporté des modifications aux fichiers Docker (comme le Dockerfile ou des dépendances), vous pouvez ajouter l'option --build pour reconstruire les images avant de démarrer les conteneurs :

docker-compose up --build

**Commandes utiles**

1. **Vérifier l'état des conteneurs** : Pour voir les conteneurs en cours d'exécution, utilisez :

docker ps

docker ps -a