Historique projet

# Installation de docker

Installé et enabled via la doc docker

<https://docs.docker.com/engine/install/centos/#install-using-the-repository>

# installation de mvn + java

<https://linuxize.com/post/how-to-install-apache-maven-on-centos-7/>

mis mvn dans le path

export PATH=/opt/apache-maven-3.8.1/bin:$PATH

# installation vers 11 de java + passé dessus (alternatives --config java)

On a décidé de partir en premier lieu sur tout en dur puis ensuite aller sur docker.

# Installation de mariadb et postgresql

<https://linuxize.com/post/install-mariadb-on-centos-7/>

<https://www.hostinger.com/tutorials/how-to-install-postgresql-on-centos-7/>

installe les 2 db sur la même machine, elle sera notre machine vitrine pour dire qu’on est capable de faire les 2.

L’idée serait de faire tourner les DB sur des dockers et l’appli aussi sur un docker.

# Faire marcher docker

Yum install wget

git clone <https://github.com/matthcol/mepdevops202106>

git clone <https://github.com/matthcol/dbmovie>

git clone https://github.com/matthcol/movieapijava2021

mvn package dans le dossier movieapijava2021

docker pull mariadb

docker pull openjdk

mkdir javamovie (dossier pour travailler)

dans mkdir :

vi Dockerfile

contient :

# syntax=docker/dockerfile:1  
FROM openjdk:latest  
COPY . .  
CMD ["java","-jar","movieapp.jar"]

docker build -t movieapp

​

docker run --name moviesapp\_container movieapp

# faire marcher container mysql

créé le fichier docker-compose.yml :

version: "3.3"

services:

database:

image: mysql:latest

container\_name: mysql\_dbmovie

volumes:

- mysql\_dbmovie\_volume:/var/lib/mysql

- ./db\_scripts:/docker-entrypoint-initdb.d

environment:

MYSQL\_ROOT\_PASSWORD: password

MYSQL\_DATABASE: dbmovie

MYSQL\_USER: movie

MYSQL\_PASSWORD: password

java:

build: .

ports:

- "5000:5000"

image: "openjdk"

volumes:

mysql\_dbmovie\_volume: {}

database🡪 le nom de notre container

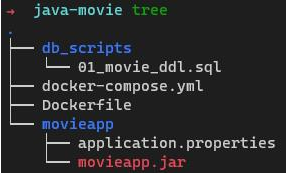
image🡪 template que docker utilise pour le créer.

Volumes🡪 le 1er volume on créé un volume qui a le nom à gauche des : et sauvegarde (du docker) ce qui est à droite des :.

2e volume copie un chemin et son contenu de notre machine hote (à gauche des : ) vers un dossier dans le container ( à droite des : )

Environnement🡪 variables utilisées par mysql (attention à bien mettre des : et non des = ; le yml lit les : ).

fait ensuite docker-compose up database



Modifié le dockerfile :

# syntax=docker/dockerfile:1  
FROM openjdk:11  
COPY ./movieapp .  
CMD ["java","-jar","movieapp.jar"]

Rajouté lignes dans le pom.xml :

<!-- https://mvnrepository.com/artifact/mysql/mysql-connector-java -->

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

</dependency>

# Jenkins

A créé le pipeline

Script pour lancer à distance le job :

curl -u admin:password <http://192.168.1.167:8080/job/movie_docker_CI/build?token=movie_jeton>

admin :password est l’utilisateur et le mdp jenkins (à mettre dans un script protégé) ; on a créé le token movie\_jeton et mis sur jenkins. L’ip est celui de anthony.