

Rapport : Dockerisation d'une Application Java RMI

Réalisé par :
Imane Hadi
Ezzaimi Salma

1. Objectif du projet

Le but de ce projet est de dockeriser une application Java basée sur RMI (Remote Method Invocation), composée d'un serveur et d'un client, en suivant les bonnes pratiques DevOps.

L'objectif final est de :

- Simplifier le déploiement
- Garantir la portabilité de l'application
- Comprendre les principes de conteneurisation avec Docker

2. Définitions importantes

- **Java RMI (Remote Method Invocation)** : Technologie Java permettant à un programme d'invoquer des méthodes sur un objet distant (généralement sur une autre machine), comme s'il était local.
- **Serveur RMI** : Programme qui expose des objets distants. Il les enregistre dans un registre RMI.
- **Client RMI** : Programme qui consulte le registre RMI pour accéder aux objets distants exposés par le serveur.
- **Docker** : Plateforme permettant d'automatiser le déploiement d'applications dans des conteneurs logiciels.
- **Conteneur Docker** : Unité légère et portable contenant tout ce qu'il faut pour exécuter une application (code, dépendances, configuration).
- **Image Docker** : Modèle utilisé pour créer des conteneurs Docker. C'est une sorte de "photo" du système de fichiers et des paramètres à un instant

donné.

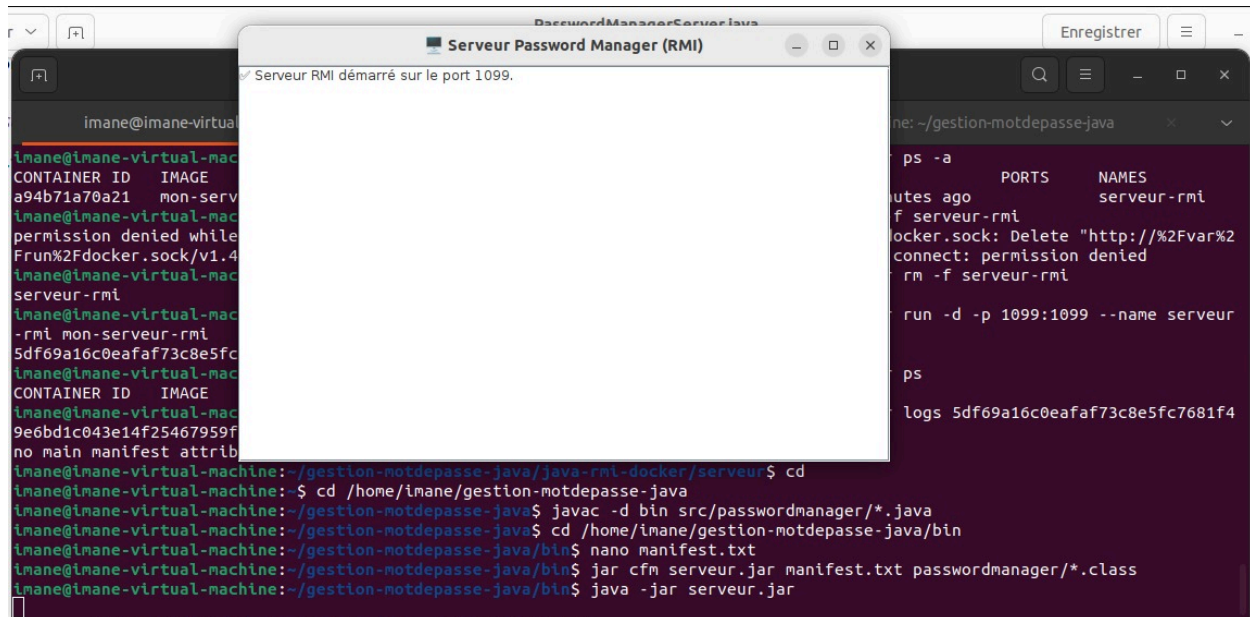
- **Dockerfile** : Fichier texte décrivant étape par étape comment construire une image Docker.
- **Docker Network** : Réseau privé Docker permettant à plusieurs conteneurs de communiquer entre eux.

3. Structure du projet

gestion-motdepasse-java/

```
|— src/
|   |— passwordmanager/
|       |— PasswordManagerClient.java
|       |— PasswordManagerInterface.java
|       |— PasswordManagerImpl.java
|       |— PasswordManagerServer.java
|— server/
|   |— Dockerfile
|   |— mon-app.jar
|— client/
|   |— Dockerfile
|   |— mon-app-client.jar
|— build-and-run.sh
|— README.md
```

4. Compilation et création des JARs



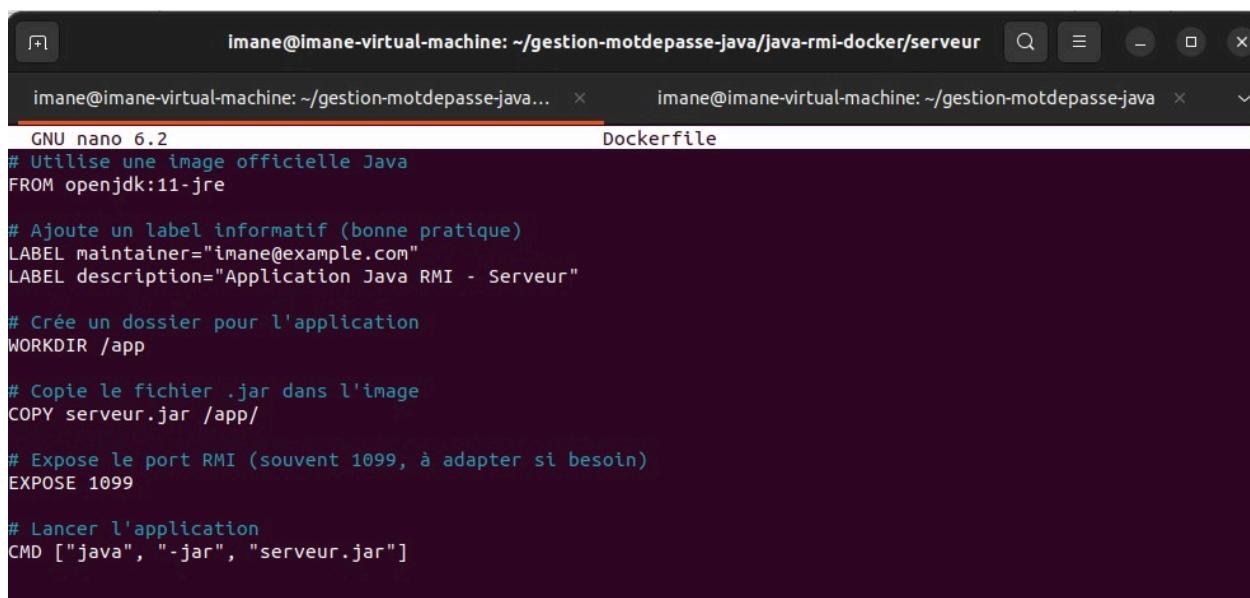
```
imane@imane-virtual-machine: ~/gestion-motdepasse-java/java-rmi-docker/serveur$ cd
imane@imane-virtual-machine: ~$ cd /home/imane/gestion-motdepasse-java
imane@imane-virtual-machine: ~/gestion-motdepasse-java$ javac -d bin src/passwordmanager/*.java
imane@imane-virtual-machine: ~/gestion-motdepasse-java$ cd /home/imane/gestion-motdepasse-java/bin
imane@imane-virtual-machine: ~/gestion-motdepasse-java/bin$ nano manifest.txt
imane@imane-virtual-machine: ~/gestion-motdepasse-java/bin$ jar cfm serveur.jar manifest.txt passwordmanager/*.class
imane@imane-virtual-machine: ~/gestion-motdepasse-java/bin$ java -jar serveur.jar
```

Serveur Password Manager (RMI)

✓ Serveur RMI démarré sur le port 1099.

```
imane@imane-virtual-machine: ~/gestion-motdepasse-java$ javac -d bin-client src/passwordmanager/*.java
imane@imane-virtual-machine: ~/gestion-motdepasse-java$ cd bin-client
imane@imane-virtual-machine: ~/gestion-motdepasse-java/bin-client$ echo "Main-Class: passwordmanager.TaClasseClient" > manifest.txt
imane@imane-virtual-machine: ~/gestion-motdepasse-java/bin-client$ jar cfm client.jar manifest.txt passwordmanager/
imane@imane-virtual-machine: ~/gestion-motdepasse-java/bin-client$ mkdir -p ~/gestion-motdepasse-java/java-rmi-docker/client
imane@imane-virtual-machine: ~/gestion-motdepasse-java/bin-client$ cp client.jar ~/gestion-motdepasse-java/java-rmi-docker/client/
imane@imane-virtual-machine: ~/gestion-motdepasse-java/bin-client$ nano Dockerfile
imane@imane-virtual-machine: ~/gestion-motdepasse-java/bin-client$
```

5. Dockerisation du serveur



```
imane@imane-virtual-machine: ~/gestion-motdepasse-java/java-rmi-docker/serveur
```

GNU nano 6.2 Dockerfile

```
# Utilise une image officielle Java
FROM openjdk:11-jre

# Ajoute un label informatif (bonne pratique)
LABEL maintainer="imane@example.com"
LABEL description="Application Java RMI - Serveur"

# Crée un dossier pour l'application
WORKDIR /app

# Copie le fichier .jar dans l'image
COPY serveur.jar /app/

# Expose le port RMI (souvent 1099, à adapter si besoin)
EXPOSE 1099

# Lancer l'application
CMD ["java", "-jar", "serveur.jar"]
```

Commandes Docker :

```
imane@imane-virtual-machine:~/gestion-motdepasse-java/java-rmi-docker/serveur$ sudo docker build -t mon-serveur-rmi .
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
             Install the buildx component to build images with BuildKit:
             https://docs.docker.com/go/buildx/

Sending build context to Docker daemon  10.75kB
Step 1/7 : FROM openjdk:11-jre
--> 362cda5d270e
Step 2/7 : LABEL maintainer="imane@example.com"
--> Using cache
--> 94df4cb9c748
Step 3/7 : LABEL description="Application Java RMI - Serveur"
--> Using cache
--> 5c78e73e8bff
Step 4/7 : WORKDIR /app
--> Using cache
--> 89da2a93a89c
Step 5/7 : COPY serveur.jar /app/
--> ba610cf1284a
Step 6/7 : EXPOSE 1099
--> Running in 98cff9150a01
--> Removed intermediate container 98cff9150a01
--> dce182b15413
Step 7/7 : CMD ["java", "-jar", "serveur.jar"]
```

```
Step 7/7 : CMD ["java", "-jar", "serveur.jar"]
--> Running in 9384e955ac58
--> Removed intermediate container 9384e955ac58
--> 9f425b96896d
Successfully built 9f425b96896d
Successfully tagged mon-serveur-rmi:latest
```

```
imane@imane-virtual-machine:~/gestion-motdepasse-java/java-rmi-docker/serveur$ sudo docker run -d -p 1099:1099 --name serveur
-rmi2 mon-serveur-rmi
70cbf848cacd99860aba8031d6ca5fd36f865ccd0fb006f62196f7cc2b463d6b
```

```
imane@imane-virtual-machine:~/gestion-motdepasse-java/java-rmi-docker/serveur$ sudo docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS          NAMES
70cbf848cacd   mon-serveur-rmi  "java -jar serveur.j..." 46 seconds ago Exited (1) 43 seconds ago           serveur-rmi2
5df69a16c0ea   5b05bf30bac6   "java -jar serveur.j..." 25 minutes ago Exited (1) 25 minutes ago           serveur-rmi
imane@imane-virtual-machine:~/gestion-motdepasse-java/java-rmi-docker/serveur$
```

6. Script d'automatisation

Fichier : [build-and-run.sh](#)

```
imane@imane-virtual-machine: ~/gestion-motdepasse-java/java-rmi-docker/serveur
GNU nano 6.2 build.sh *
#!/bin/bash

echo "🔨 Construction de l'image Docker..."
docker build -t mon-serveur-rmi .

if [ $? -eq 0 ]; then
    echo "✅ Image construite avec succès : mon-serveur-rmi"
else
    echo "❌ Erreur lors de la construction de l'image"
    exit 1
fi

appendLog("❌ Erreur serveur : " + e.getMessage());
```

```
imane@imane-virtual-machine: ~/gestion-motdepasse-java/java-rmi-docker/serveur
GNU nano 6.2 run.sh *
#!/bin/bash

echo "🚀 Lancement du conteneur Docker..."

# Vérifie si un conteneur avec le même nom existe déjà
if [ "$(docker ps -aq -f name=serveur-rmi)" ]; then
    echo "🗑 Suppression de l'ancien conteneur..."
    docker rm -f serveur-rmi
fi

# Lance un nouveau conteneur
docker run -d -p 1099:1099 --name serveur-rmi mon-serveur-rmi

if [ $? -eq 0 ]; then
    echo "✅ Conteneur lancé avec succès sur le port 1099"
else
    echo "❌ Erreur lors du lancement du conteneur"
fi

appendLog("❌ Erreur serveur : " + e.getMessage());
```

7. Dockerisation du client

```

imane@imane-virtual-machine:~/gestion-motdepasse-java/java-rmi-docker/client$ nano Dockerfile
imane@imane-virtual-machine:~/gestion-motdepasse-java/java-rmi-docker/client$ sudo docker build -t mon-client-rmi .
[sudo] Mot de passe de imane :
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
             Install the buildx component to build images with BuildKit:
             https://docs.docker.com/go/buildx/

Sending build context to Docker daemon  10.75kB
Step 1/4 : FROM openjdk:17-jdk-slim
--> 37cb44321d04
Step 2/4 : WORKDIR /app
--> Using cache
--> 470c0bdb661f
Step 3/4 : COPY client.jar /app/
--> 98bd02ab902f
Step 4/4 : CMD ["java", "-jar", "client.jar"]
--> Running in ccb5b56ce561
--> Removed intermediate container ccb5b56ce561
--> 45bd705bd527
Successfully built 45bd705bd527
Successfully tagged mon-client-rmi:latest
imane@imane-virtual-machine:~/gestion-motdepasse-java/java-rmi-docker/client$

```

8. Résultats obtenus

```

imane@imane-virtual-machine:~/gestion-motdepasse-java/java-rmi-docker/serveur$ sudo ./run.sh
🚀 Lancement du conteneur Docker...
🔧 Suppression de l'ancien conteneur...
serveur-rmi
8968a01e445d8b8f8566aace3f3feea9e3579467b38bb12a02f2da6a6f8af7c0
✅ Conteneur lancé avec succès sur le port 1099

```