Réalisation d'un Gestionnaire de Mots de Passe avec RPC et Docker

Abdelmouiz bensbai / Reda hamdi

June 17, 2025

Contents

Chapter 1

Introduction

Ce rapport présente la réalisation d'un gestionnaire de mots de passe en Java, utilisant la communication RPC (Remote Procedure Call) et la conteneurisation avec Docker. L'objectif est de développer une application sécurisée, portable et accessible via une interface graphique, même dans un environnement conteneurisé.

Chapter 2

Architecture du Projet

Le projet est composé de deux parties principales :

- **Serveur RPC** : Gère la logique métier et la gestion sécurisée des mots de passe.
- **Client RPC** : Fournit une interface graphique à l'utilisateur pour interagir avec le serveur.

La communication entre le client et le serveur se fait via RPC, permettant l'appel de méthodes distantes de manière transparente.

2.1 Schéma d'architecture

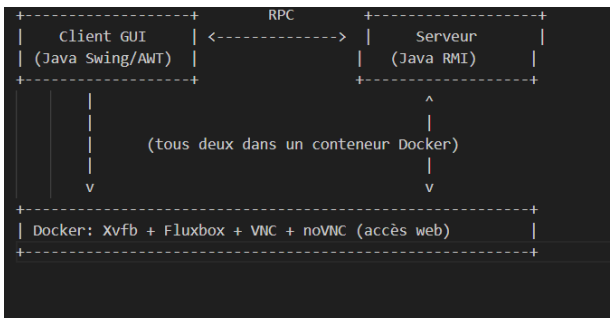

Figure 2.1: Schéma d'architecture du projet

Chapter 3

Utilisation de Docker

Docker a été utilisé pour garantir la portabilité et l'isolation de l'application. Grâce à Docker, il est possible d'exécuter l'application sur n'importe quelle machine sans se soucier des dépendances système.

3.1 Concepts Docker

- **Image Docker** : Un snapshot contenant tout le nécessaire pour exécuter l'application (Java, dépendances, code source).
- **Conteneur Docker** : Instance en cours d'exécution d'une image.
- **Dockerfile** : Fichier de configuration pour construire l'image.
- **Volumes** : Permettent de partager des fichiers entre l'hôte et le conteneur.
- **Ports** : Permettent d'exposer des services du conteneur vers l'extérieur (ex: noVNC sur le port 8080).

3.2 Commandes principales utilisées

```
# Construction de l'image
sudo docker build -t rpc-password-manager .

# Lancement du conteneur avec acc s web    l'interface graphique
sudo docker run -d --name rpc-app -p 8080:8080 -v ~/Desktop/rpc_appp:/app r
```

3.3 Accès à l'interface graphique via le web

L'accès à l'interface graphique s'est fait via noVNC, accessible à l'adresse `http://localhost:8080/vnc.html`.

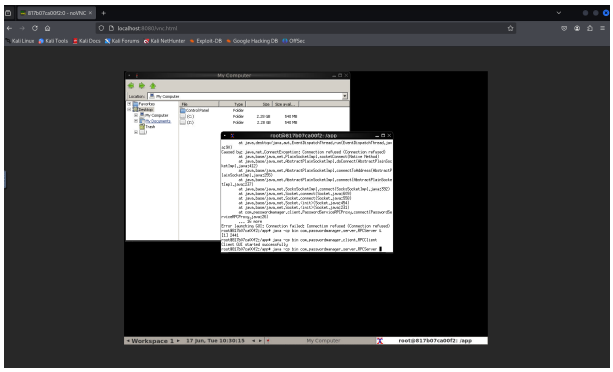

Figure 3.1: Interface Docker/noVNC

Chapter 4

Utilisation de RPC

Le protocole RPC permet au client d'appeler des méthodes distantes sur le serveur comme s'il s'agissait de méthodes locales. Cela facilite la séparation des responsabilités et la modularité du code.

4.1 Exemple de code RPC

```
PasswordService service = (PasswordService) Naming.lookup("//localhost/Pass
service.addPassword("site", "username", "password");
```

Chapter 5

Interface graphique

5.1 Interface Client

5.2 Interface Serveur

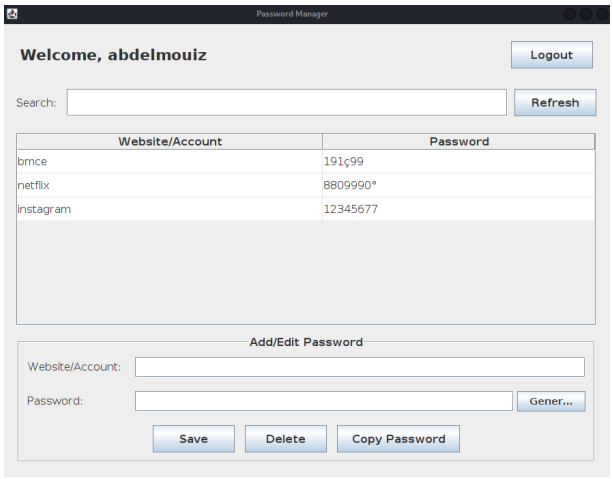

Figure 5.1: Interface client

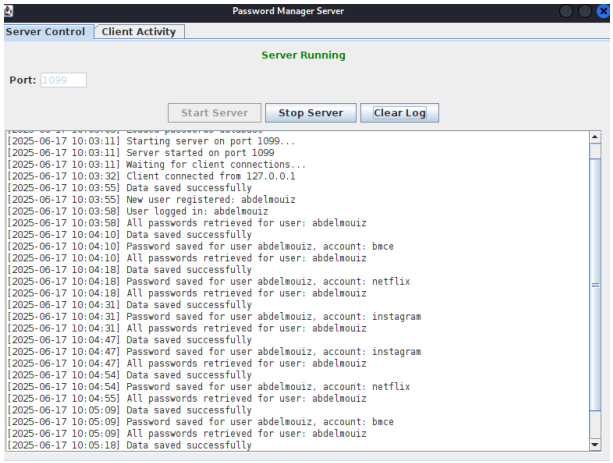

Figure 5.2: Interface serveur

Chapter 6

Conclusion

Ce projet a permis de mettre en pratique la communication RPC en Java et la conteneurisation avec Docker, tout en assurant l'accès à une interface graphique via le web grâce à noVNC. Cette approche garantit portabilité, sécurité et simplicité de déploiement.