

## Manuel d'installation et d'utilisation – SAE302

### Architecture du système

Le système repose sur un scanner Java, une base de données MariaDB, une API PHP, un site web et une application Android. Le scénario retenu repose sur une machine physique hébergeant l'application Android, et une machine virtuelle Linux destinée à accueillir l'ensemble des composants serveur, la base de données, le site web et le scanner Java.

### Projet

- Téléchargement de l'archive SAE302.zip contenant SAE32Scanner, sae302android, SiteWeb et sae302.sql
- Extraction du projet : `unzip SAE302.zip`

### Dans la VM

#### Base de données

- Installation : `sudo apt update` puis `sudo apt install mariadb-server mariadb-client -y`
- Démarrage du service : `sudo systemctl enable mariadb` et `sudo systemctl start mariadb`
- Import de la base : `sudo mariadb < sae302.sql`

#### Site web et API

- Installer Apache et PHP : `sudo apt install apache2 php libapache2-mod-php -y`
- Placer les fichiers PHP du projet dans le répertoire `/var/www/html`
- Vérifier les droits d'accès si nécessaire
- Démarrer et activer Apache : `sudo systemctl start apache2` et `sudo systemctl enable apache2`
- Accéder au site web et à l'API via <http://localhost/site.php>

#### Ajout d'un scan

- Connexion à la base de données : `sudo mariadb` puis `USE sae302`
- Insertion d'un nouveau scan avec le statut pending : `INSERT INTO scanner_runs (target, status) VALUES ('192.168.56.102', 'pending');`
- Le scanner Java détecte le scan en attente et lance l'analyse
- Les résultats sont stockés en base et le statut est mis à jour

## Scanner JAVA

- Installer le Java JDK : *sudo apt install default-jdk -y*
- Installer le connecteur MariaDB : *sudo apt install libmariadb-java -y*
- Se placer dans le répertoire racine du scanner
- Compiler les sources : *javac com/sae302/scanner/\*.java*
- Exécuter le scanner avec le classpath JDBC : *sudo java -cp .:/usr/share/java/mariadb-java-client.jar com.sae302.scanner.ScanApp*
- Le scanner effectue l'analyse et enregistre les résultats dans la base de données

## Redirection de port (Port Forwarding)

- L'API étant hébergée sur une machine virtuelle en mode NAT, une redirection de port est nécessaire afin de permettre l'accès depuis l'application Android exécutée sur la machine hôte.
- Sélectionner la machine virtuelle dans le VirtualBox
- Cliquer sur Configuration → Réseau → Carte 1
- Vérifier que le mode réseau est NAT
- Cliquer sur Avancé → Redirection de ports
- Ajouter une règle avec les paramètres suivants :
  - Protocole : TCP
  - Port hôte : 8080
  - Port invité : 80
- Valider et démarrer la machine virtuelle
- Vérifier que l'API est accessible depuis la machine hôte via un navigateur : <http://localhost:8080/site.php>

## En dehors de la VM

### Application Android

- Ouvrir le projet dans Android Studio
- Ouvrir le fichier MainActivity.java
- Configurer l'URL de base de l'API à la ligne : *private static final String API\_BASE = "http://10.0.2.2:8080/api/failles";*
- L'adresse 10.0.2.2 permet à l'émulateur Android d'accéder à la machine hôte
- Le port et le chemin correspondent à l'API PHP hébergée par Apache
- Compiler et lancer l'application sur un émulateur ou un appareil Android
- L'application récupère les données depuis l'API et affiche les résultats des scans

### Utilisation

L'administrateur lance les scans, les utilisateurs consultent les résultats via le site web ou l'application Android.