

Document d'installation

Projet SAE 3.02 – Routage en oignon

Nom du groupe: π Thon

Membres du groupe:

- HENRY Aurélien
- HALTER Mathis

Table des matières

1.Prérequis	1
2.Récupération du projet	2
Méthode 1 : via GitHub (recommandée)	2
Méthode 2 : téléchargement manuel	2
3.Installation de Python	2
4.Installation de MariaDB	3
5.Installation des bibliothèques Python	3
6.Organisation de l'installation	3
7.Schéma de l'architecture	4

1.Prérequis

Le projet peut être installé et utilisé sur une ou plusieurs machines.

Systèmes d'exploitation compatibles :

- Linux
- Windows
- MacOS

Logiciels requis :

- Python 3.10 ou supérieur
- Maria DB
- Git (optionnel)

2. Récupération du projet

Les sources du projet sont disponibles sur GitHub dans le dossier Sources.

Deux méthodes sont possibles :

Méthode 1 : via GitHub (recommandée)

```
git clone <URL_DU_DEPOT_GITHUB>
```

Puis récupérer le dossier Sources_codes_final.

Méthode 2 : téléchargement manuel

Télécharger l'archive ZIP du projet depuis GitHub

Extraire l'archive

Récupérer le dossier Sources_codes_final

3. Installation de Python

Si Python n'est pas installé, suivre les instructions ci-dessous.

Sous Linux (Debian / Ubuntu)

```
sudo apt update
```

```
sudo apt install python3 python3-pip
```

Vérification :

```
python3 --version
```

Sous Windows

Télécharger Python depuis : <https://www.python.org/downloads/>

Lancer l'installateur

Cocher "Add Python to PATH"

Finaliser l'installation

Vérification (Invite de commandes) :

```
python --version
```

Sous macOS

Avec Homebrew :

```
brew install python
```

Vérification :

```
python3 --version
```

4. Installation de MariaDB

Sous Linux (Debian / Ubuntu)

```
sudo apt install mariadb-server
```

```
sudo systemctl start mariadb
```

```
sudo systemctl enable mariadb
```

Sous Windows

Télécharger l'installateur MariaDB : <https://mariadb.org/download/>

Installer avec les options par défaut et renseigner le mot de passe root que vous voulez (il faudra le renseigner dans la commande de lancement du master)

Sous macOS

```
brew install mariadb
```

```
brew services start mariadb
```

5. Installation des bibliothèques Python

Toutes les dépendances nécessaires au fonctionnement du projet sont regroupées dans le fichier dependances.py fourni dans le dépôt GitHub.

5.1 Installation automatique

Depuis le dossier Sources_codes_final, exécuter le script correspondant à votre système :

Windows :

Double-cliquer sur installer_dependances_WIN.bat ou exécuter dans un terminal PowerShell :

```
installer_dependances_WIN.bat
```

Linux :

Dans un terminal :

```
chmod +x installer_dependances_LINUX.sh
```

```
./installer_dependances_LINUX.sh
```

macOS :

Dans un terminal :

```
./installer_dependances_MAC.sh
```

Ces scripts détectent automatiquement le système et lancent le script Python `dependances.py`, qui gère l'installation des bibliothèques via `apt`, `pip` ou `Homebrew`, selon le système.

5.2 Bibliothèques principales utilisées

La majorité des bibliothèques utilisées sont des bibliothèques standards de Python.

Les principales sont :

- `random` : génération aléatoire, choix de routes,
- `math` : opérations mathématiques pour le chiffrement,
- `socket` : communication réseau TCP/IP,
- `threading` : gestion des connexions simultanées,
- `time` : temporisation et gestion des délais,
- `sys` : gestion des arguments de la ligne de commande,
- `PyQt5` : création de l'interface graphique (Master, Clients),
- `mysql.connector` : connexion et manipulation de la base de données MariaDB.

5.3 Vérification de l'installation

Après l'exécution du script d'installation, vous pouvez vérifier que toutes les dépendances sont correctement installées avec les commandes suivantes :

```
python3 -c "import PyQt5; print('PyQt5 OK')"
```

```
python3 -c "import mysql.connector; print('MySQL OK')"
```

Si vous obtenez :

```
PyQt5 OK
```

```
MySQL OK
```

alors toutes les bibliothèques sont installées et le projet est prêt à l'exécution.

5.4 Remarques importantes

- Sur Linux Kali, le script utilise l'option `--break-system-packages` pour pip afin de contourner la protection PEP 668. Cela est nécessaire pour installer `mysql-connector-python` directement dans le système sans `virtualenv`.
- L'installation peut prendre plusieurs dizaines de secondes lors de la première exécution, notamment pour `mysql-connector-python` (~34 Mo). Les installations suivantes seront beaucoup plus rapides.
- Le projet fonctionne de manière uniforme sur Windows, Linux et macOS sans nécessiter de modification manuelle.

6. Organisation de l'installation

Les composants du projet (Master, routeurs et clients) peuvent être installés sur une seule machine ou sur plusieurs machines distinctes.

Le nombre de routeurs et de clients pouvant être installés n'est pas limité.

7. Schéma de l'architecture

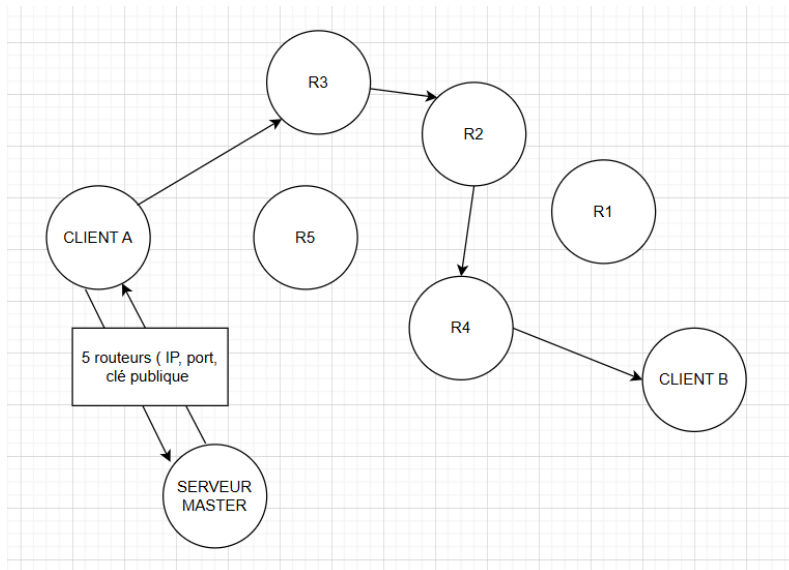


Schéma pour 5 routeurs