



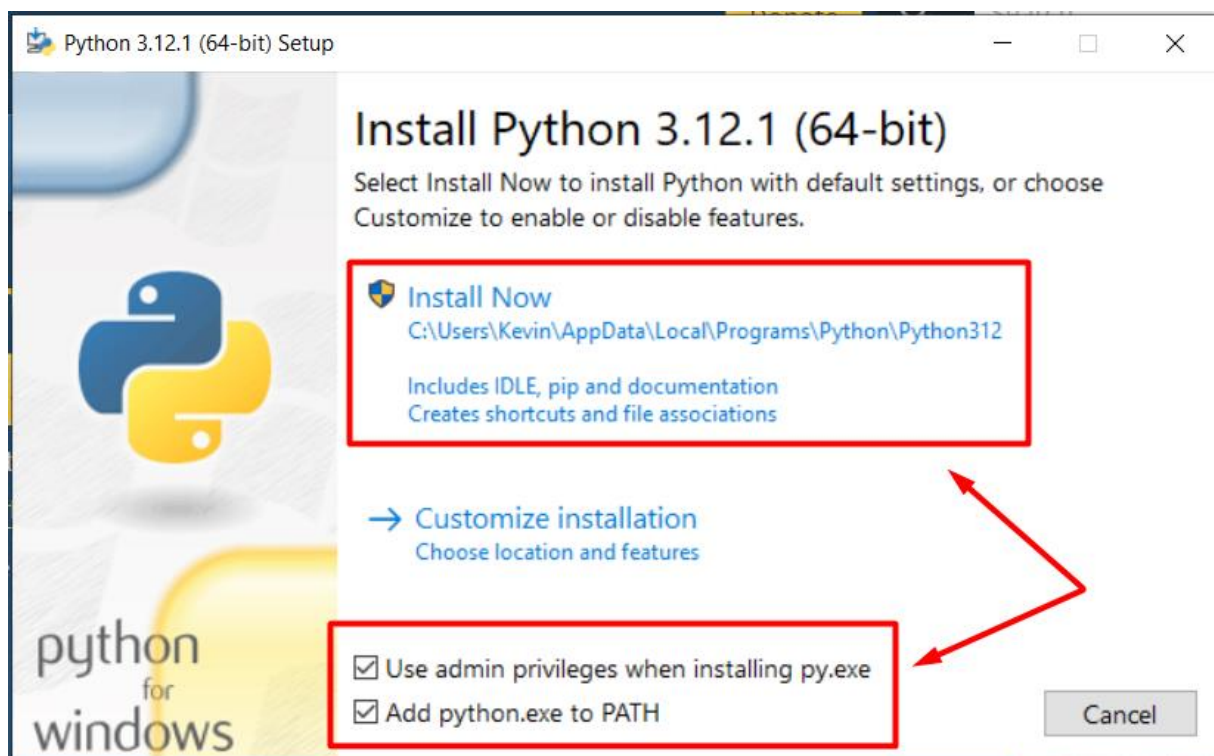
→ Pour cette procédure d'installation, on va considérer que le serveur ainsi que les clients seront installés sur des environnements standards Windows.

## I – Installation du serveur :

1. On télécharge la version 3.12 de Python ici :

<https://www.python.org/ftp/python/3.12.1/python-3.12.1-amd64.exe>

2. On lance l'exécutable téléchargé en prenant soin, au cours de l'installation, de cocher la case « **Add Python to PATH** » !



3. Une fois l'installation terminée, on lance un terminal PowerShell et on entre la commande suivante pour nous rendre au répertoire d'installation par défaut de Python :

« **cd .\AppData\Local\Programs\Python\Python312\** »

```
Sélection Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. Tous droits réservés.

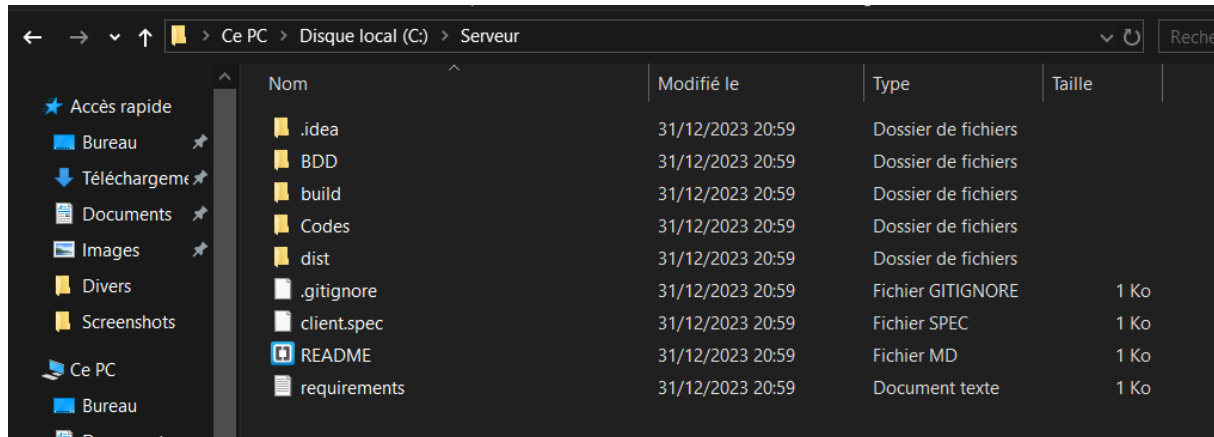
Testez le nouveau système multiplateforme PowerShell https://aka.ms/pscore6

PS C:\Users\Kevin> cd .\AppData\Local\Programs\Python\Python312\
PS C:\Users\Kevin\AppData\Local\Programs\Python\Python312>
```

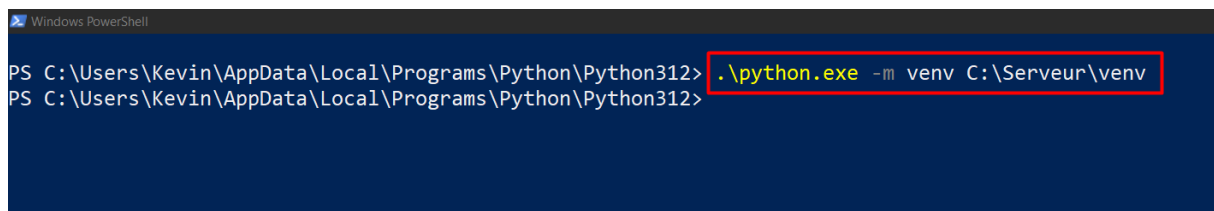
4. On laisse notre fenêtre PowerShell de côté pendant que nous allons désormais télécharger le fichier .zip contenant le projet, sur ce lien :

<https://github.com/Kevin-D-A/SAE-3.02/archive/refs/heads/main.zip>

5. Une fois le fichier téléchargé, on l'extrait dans un dossier Serveur situé à la racine du disque C, comme suit :



6. Nous allons désormais créer un environnement virtuel dans ce dossier Serveur. Pour cela, on retourne dans notre terminal PowerShell et on exécute la commande suivante : « `.\python.exe -m venv C:\Serveur\venv` »



7. Il nous faudra ensuite activer l'environnement virtuel, à l'aide de la commande suivante : « `C:\Serveur\venv\Scripts\activate` »

```
PS C:\Users\Kevin\AppData\Local\Programs\Python\Python312> C:\Serveur\venv\Scripts\activate
```

8. Cela fait, on se rend dans le dossier de notre environnement virtuel à l'aide de la commande : « `cd C:\Serveur\venv\Scripts\` »

```
PS C:\Users\Kevin\AppData\Local\Programs\Python\Python312> cd C:\Serveur\venv\Scripts\
```

9. Puis on installe les dépendances requises par notre programme serveur avec la commande suivante :

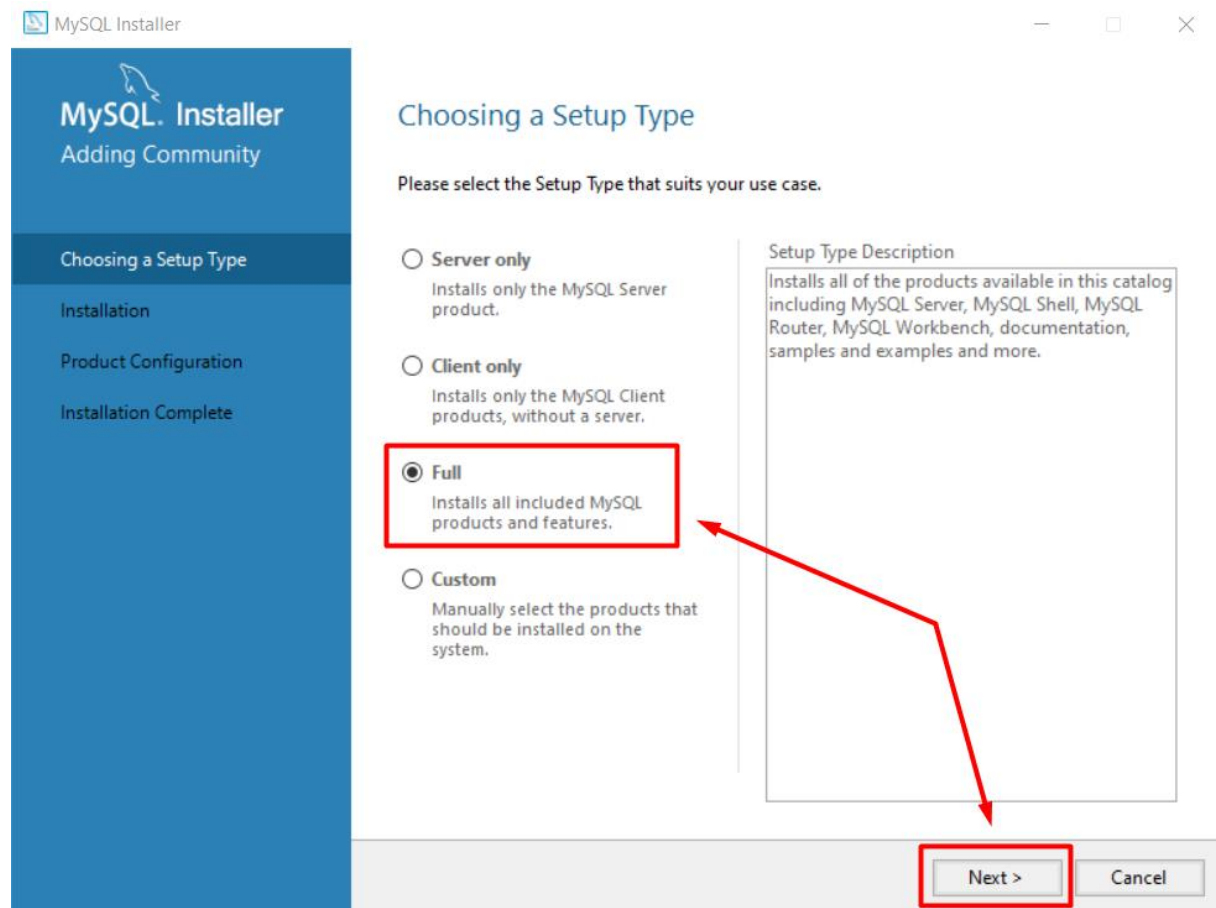
« `pip install -r C:\Serveur\requirements.txt` »

```
PS C:\Serveur\venv\Scripts> pip install -r C:\Serveur\requirements.txt
```

10. Une fois l'installation des dépendances terminée, on conserve une nouvelle fois notre terminal PowerShell ouvert, pendant que nous allons installer MySQL, dont voici le lien :

<https://dev.mysql.com/get/Downloads/MySQLInstaller/mysql-installer-community-8.0.35.0.msi>

11. Le fichier téléchargé, on l'exécute et sur la première fenêtre, on coche « **Full** » puis on clique sur « **Next** » :



12. On cliquera ensuite sur le bouton « **Execute** » puis sur les prochains boutons « **Next** » en laissant les valeurs par défaut jusqu'à ce que l'on tombe sur la fenêtre suivante, où il nous faudra définir un mot de passe pour le compte root :

MySQL Installer

MySQL Server 8.0.35

Type and Networking

Authentication Method

**Accounts and Roles**

Windows Service

Server File Permissions

Apply Configuration

### Accounts and Roles

Root Account Password

Enter the password for the root account. Please remember to store this password in a secure place.

MySQL Root Password: .....

Repeat Password: .....

Password strength: **Weak**

### MySQL User Accounts

Create MySQL user accounts for your users and applications. Assign a role to the user that consists of a set of privileges.

MySQL User Name	Host	User Role
-----------------	------	-----------

Add User

Edit User

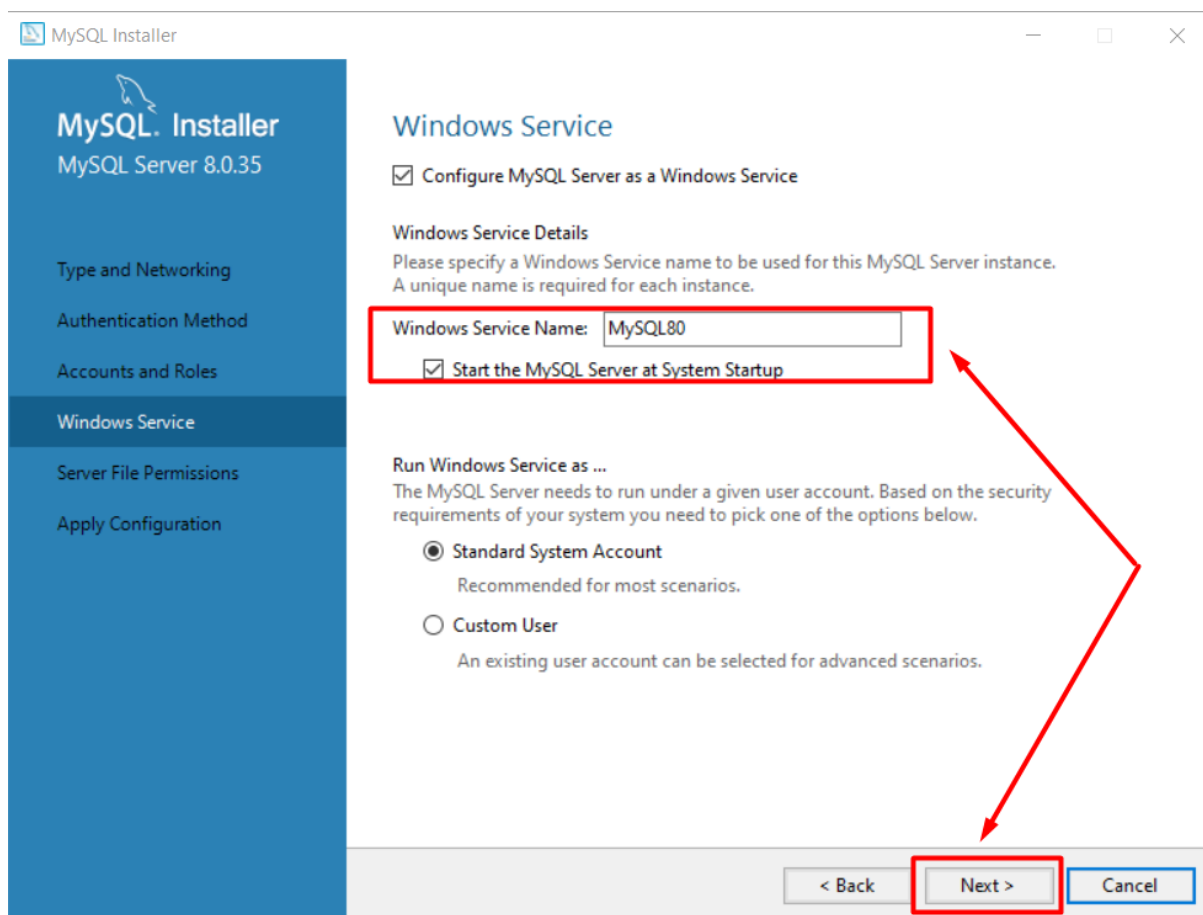
Delete

< Back

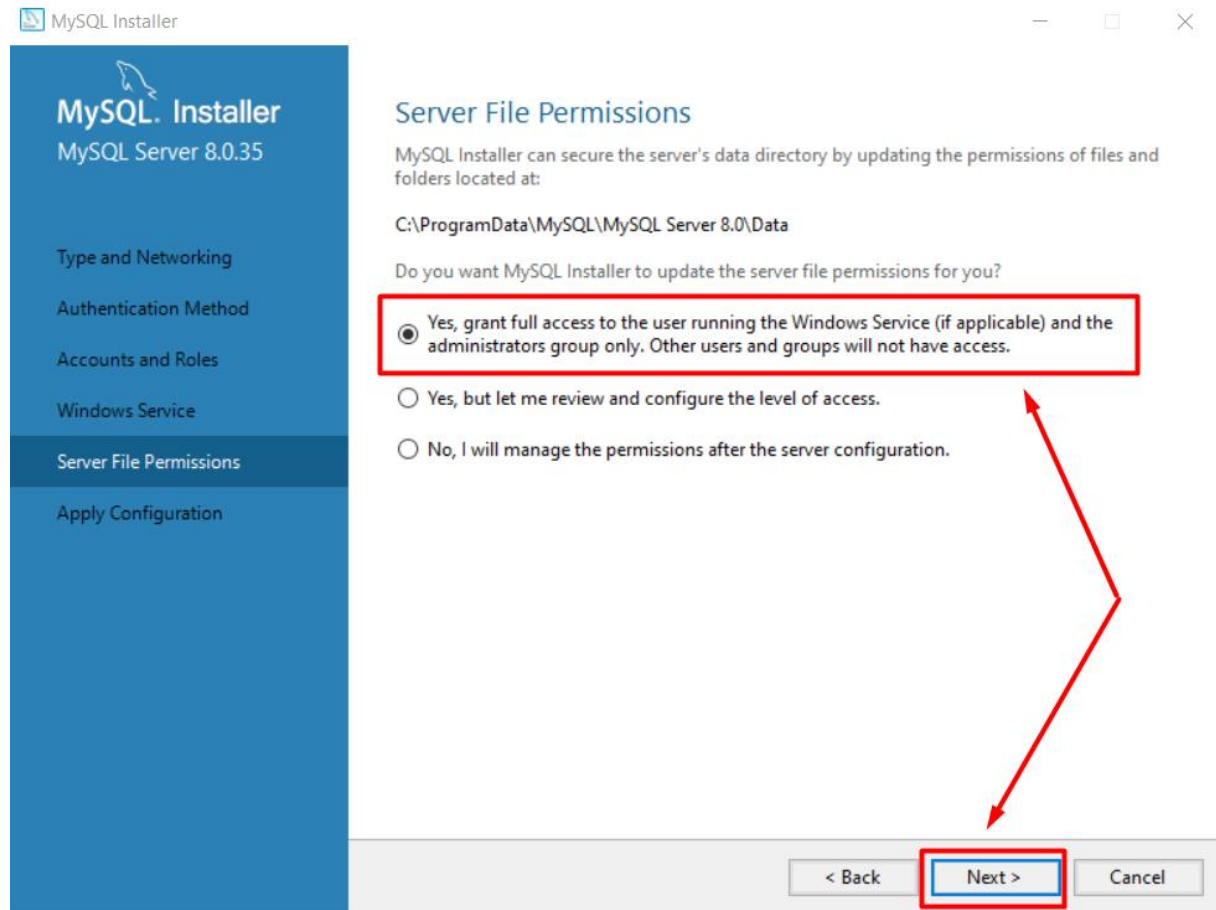
**Next >**

Cancel

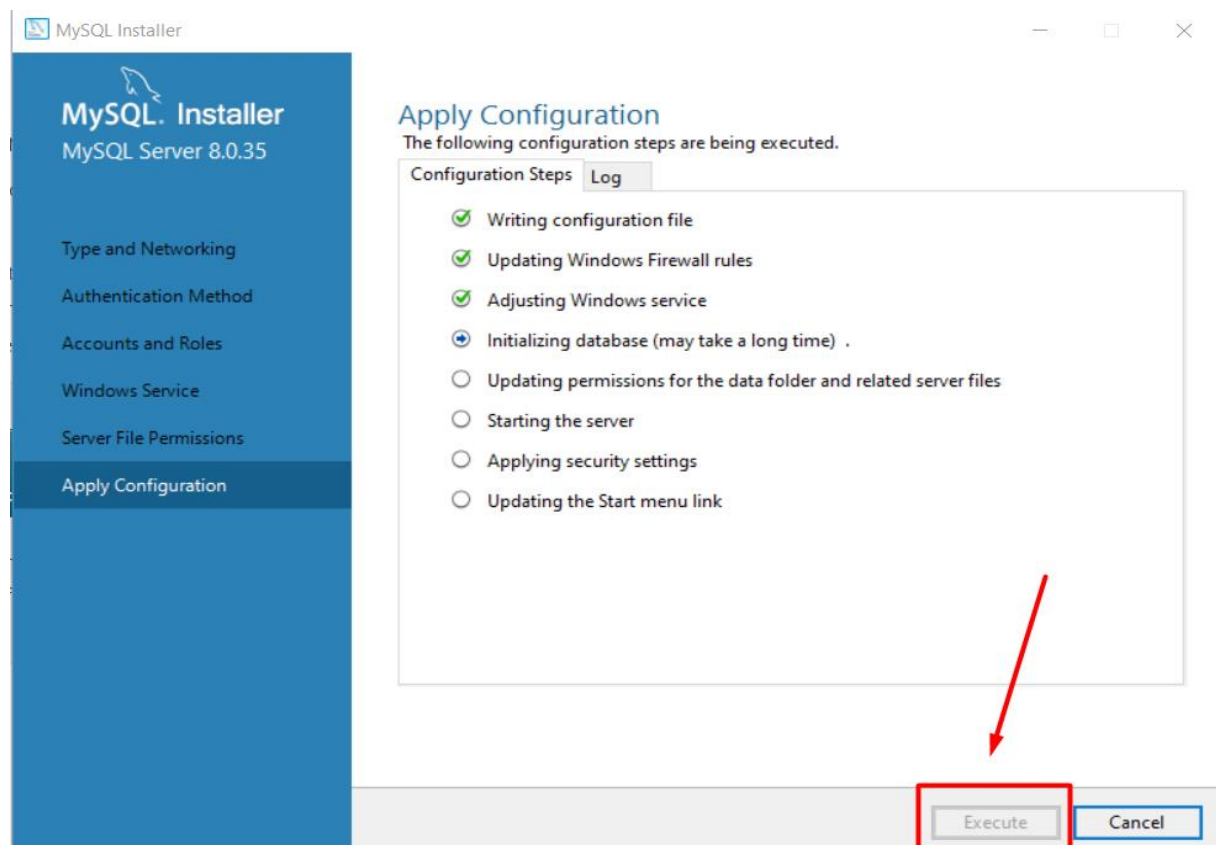
13. Une fois le mot de passe minutieusement sélectionné et conservé (robuste de préférence, pas comme le nôtre !), nous cliquerons sur « **Next** » puis, dans la prochaine fenêtre, nous nous assurons que le démarrage automatique du service MySQL est coché :



14. Nous cliquons encore sur le bouton « **Next** » ici ainsi que sur la prochaine fenêtre, en nous assurant de donner les droits complets au service :

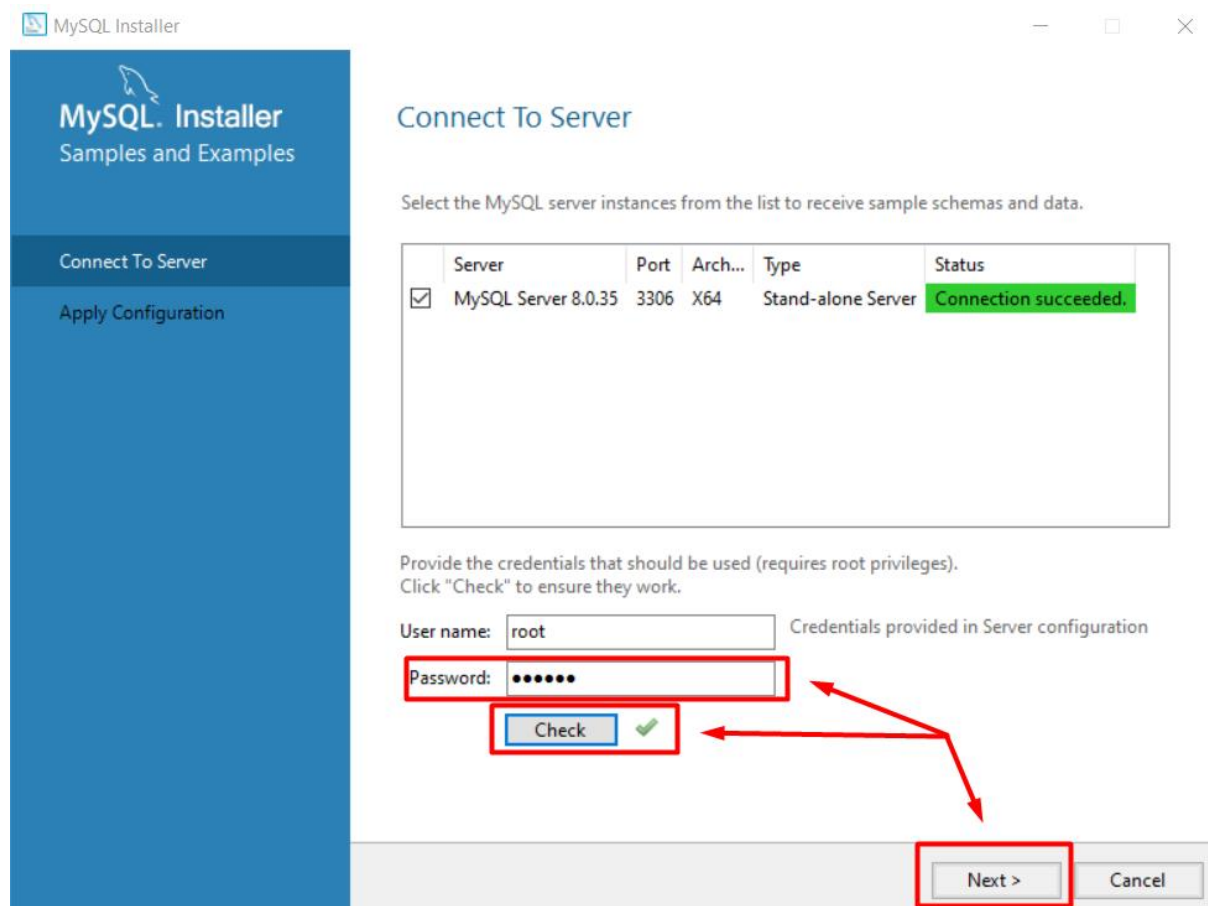


15. On cliquera sur le bouton « **Execute** » sur la prochaine fenêtre afin d'appliquer les configurations :



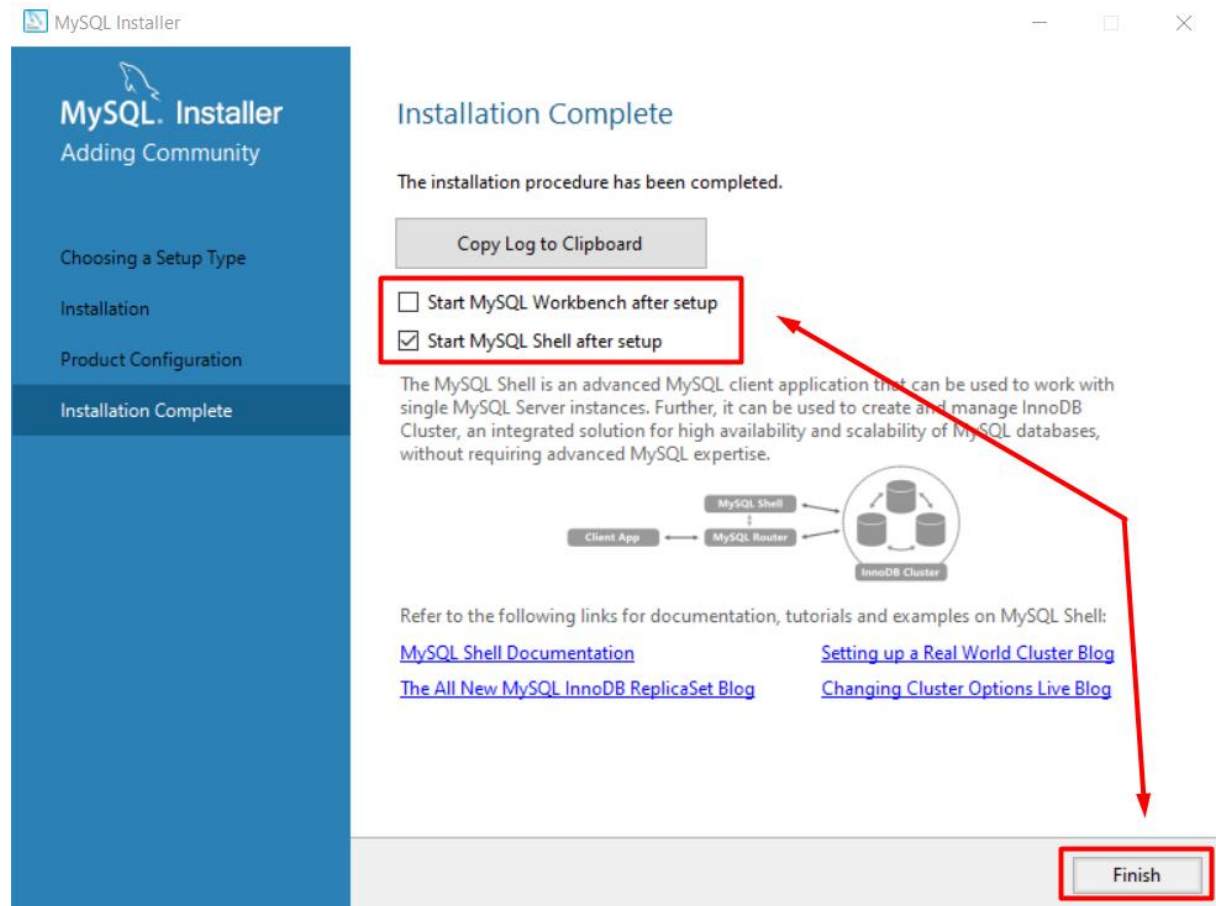


16. Une fois les configurations appliquées, on clique sur « **Finish** » et continue en laissant les valeurs par défaut jusqu'à tomber sur la fenêtre suivante, où il nous sera demandé d'entrer le mot de passe sélectionné plus tôt avant d'appuyer sur le bouton « **Check** » pour valider la connexion et enfin sur le bouton « **Next** » :





17. Dans la fenêtre suivante, on clique sur « **Execute** » puis on continue jusqu'à la dernière fenêtre, où on prendra soin de décocher « **Start MySQL Workbench after setup** » et de cocher « **Start MySQL Shell after setup** », puis enfin sur le bouton « **Finish** » :



18. Le shell MySQL s'étant désormais lancé, il nous faut mettre en place notre base de données. Pour cela, nous commençons par entrer les requêtes suivantes, comme dans l'image :

```

MySQL Shell 8.0.35

Copyright (c) 2016, 2023, Oracle and/or its affiliates.
Oracle is a registered trademark of Oracle Corporation and/or its affiliates.
Other names may be trademarks of their respective owners.

Type '\help' or '\?' for help; '\quit' to exit.
MySQL> JS> \sql
Switching to SQL mode... Commands end with ;
MySQL> SQL> \connect root@localhost:3306
Creating a session to 'root@localhost:3306'
Please provide the password for 'root@localhost:3306': *****
Save password for 'root@localhost:3306'? [Y]es/[N]o/[e]ver (default No):
Fetching global names for auto-completion... Press ^C to stop.
Your MySQL connection id is 19
Server version: 8.0.35 MySQL Community Server - GPL
No default schema selected; type \use <schema> to set one.
MySQL> localhost:3306 ssl SQL> CREATE DATABASE sae_302;
Query OK, 1 row affected (0.0120 sec)
MySQL> localhost:3306 ssl SQL> USE sae_302;
Default schema set to 'sae_302'.
Fetching global names, object names from 'sae_302' for auto-completion... Press ^C to stop.
MySQL> localhost:3306 ssl sae_302 SQL> CREATE USER 'serveur'@'localhost' IDENTIFIED BY 'sae302';
Query OK, 0 rows affected (0.0224 sec)
MySQL> localhost:3306 ssl sae_302 SQL> GRANT ALL PRIVILEGES ON sae_302.* TO 'serveur'@'localhost';
Query OK, 0 rows affected (0.0106 sec)
MySQL> localhost:3306 ssl sae_302 SQL> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.0104 sec)

```

```

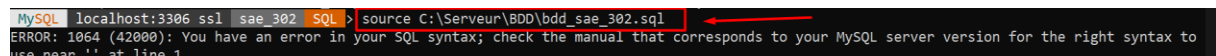
« \sql »
« \connect root@localhost:3306 »
« CREATE DATABASE sae_302 ; »
« USE sae_302 ; »
« CREATE USER 'serveur'@'localhost' IDENTIFIED BY 'sae302'; »
« GRANT ALL PRIVILEGES ON sae_302.* TO 'serveur'@'localhost'; »
« FLUSH PRIVILEGES; »

```

19. Nous avons désormais notre base de données créée et un utilisateur ayant tous les droits dessus qui permettra à notre serveur de s'y connecter. Il ne manque plus qu'à construire les tables, pour cela deux options :

- soit on effectue la commande suivante permettant d'importer le fichier SQL :

```
« source C:\Serveur\BDD\bdd_sae_302.sql ; »
```



- soit, comme dans notre cas nous avons rencontré une erreur lors de l'importation, nous allons manuellement créer les tables à l'aide des sept requêtes suivantes :

```

CREATE TABLE `clients` (
  `id_client` int NOT NULL AUTO_INCREMENT,
  `nom` varchar(255) NOT NULL,
  `prenom` varchar(255) NOT NULL,
  `email` varchar(255) NOT NULL,
  `mot_de_passe` varchar(255) NOT NULL,
  `permission` varchar(255) NOT NULL,
  PRIMARY KEY (`id_client`));

```

```

CREATE TABLE `salons_publics` (
  `id_salon_public` int NOT NULL AUTO_INCREMENT,
  `nom_salon` varchar(255) NOT NULL,
  `description` varchar(255) NOT NULL,
  PRIMARY KEY (`id_salon_public`));

```

```

CREATE TABLE `salons_prives` (
  `id_salon_prive` int NOT NULL AUTO_INCREMENT,
  `email_participant_1` varchar(255) NOT NULL,
  `email_participant_2` varchar(255) NOT NULL,
  PRIMARY KEY (`id_salon_prive`));

```

```
CREATE TABLE `messages` (
  `id_message` int NOT NULL AUTO_INCREMENT,
  `id_client` int NOT NULL,
  `contenu` text NOT NULL,
  `horodatage` datetime NOT NULL,
  `id_salon_public` int DEFAULT NULL,
  `id_salon_privé` int DEFAULT NULL,
  PRIMARY KEY (`id_message`),
  FOREIGN KEY (`id_client`) REFERENCES `clients` (`id_client`),
  FOREIGN KEY (`id_salon_privé`) REFERENCES `salons_privés` (`id_salon_privé`),
  FOREIGN KEY (`id_salon_public`) REFERENCES `salons_publics` (`id_salon_public`));
```

```
CREATE TABLE `membres_salons_publics` (
  `id_membre` int NOT NULL AUTO_INCREMENT,
  `id_client` int NOT NULL,
  `id_salon_public` int NOT NULL,
  PRIMARY KEY (`id_membre`),
  KEY `id_client` (`id_client`),
  KEY `id_salon_public` (`id_salon_public`),
  FOREIGN KEY (`id_client`) REFERENCES `clients` (`id_client`),
  FOREIGN KEY (`id_salon_public`) REFERENCES `salons_publics` (`id_salon_public`));
```

```
CREATE TABLE `historique_ip` (
  `id_historique` int NOT NULL AUTO_INCREMENT,
  `ip_client` varchar(16) NOT NULL,
  `email_client` varchar(255) NOT NULL,
  `horodatage_connexion` datetime DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (`id_historique`));
```

```
CREATE TABLE `sanctions` (
  `id_sanction` int NOT NULL AUTO_INCREMENT,
  `type_sanction` enum('ban','kick','mute') NOT NULL,
  `duree_sanction` int DEFAULT NULL,
  `motif_sanction` text NOT NULL,
  `ip_client` varchar(16) NOT NULL,
  `email_client` varchar(255) NOT NULL,
  `horodatage_sanction` datetime DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (`id_sanction`));
```

Puis ensuite nous allons créer l'utilisateur de privilège administrateur qui sera utilisé pour les commandes serveur à l'aide de la requête suivante :

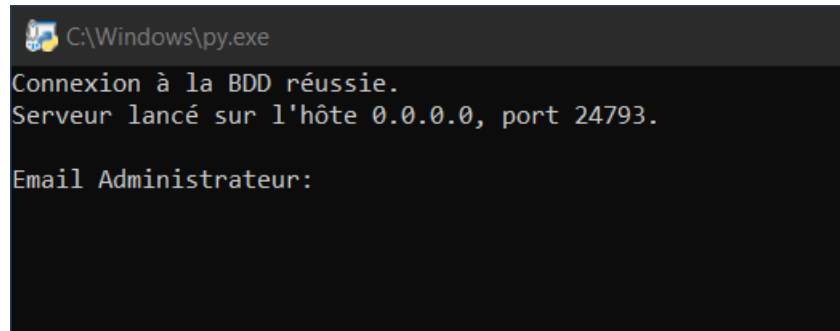
```
INSERT INTO clients (nom, prenom, email, mot_de_passe, permission)
VALUES ('admin', 'admin', 'admin@admin.com', 'admin', 'administrateur');
```

20. À ce stade, tout est en place afin de permettre au serveur de se connecter sur la base de données et d'être fonctionnel. Essayons de le lancer pour voir ! On exécute la commande suivante :

« C:\Serveur\Codes\serveur.py »

```
PS C:\Serveur\venv\Scripts> C:\Serveur\Codes\serveur.py
```

Un terminal serveur devrait alors s'ouvrir comme suit, demandant une authentification administrateur :



```
C:\Windows\py.exe
Connexion à la BDD réussie.
Serveur lancé sur l'hôte 0.0.0.0, port 24793.

Email Administrateur:
```

Nous pouvons fermer la fenêtre et passer à l'installation client.

## II - Installation du client :

Pour avoir accès à notre client, nous avons deux options.

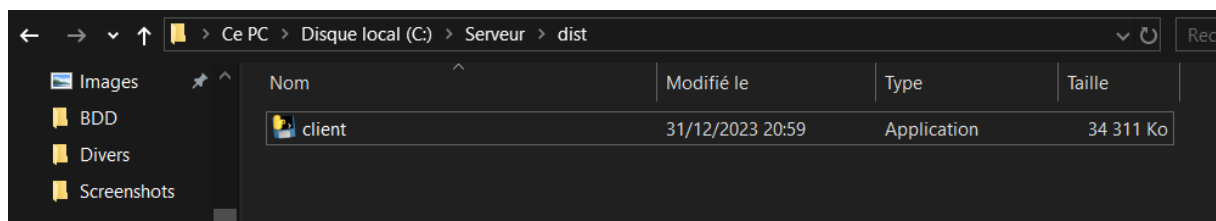
1. La première, faramineuse, consiste à répéter les étapes 1 à 9 de l'installation serveur afin de mettre en place l'environnement virtuel et les dépendances requises à l'exécution du client.

Le client pourra ensuite être exécuté avec cette commande (supposant que les mêmes répertoires sont utilisés) :

« `C:\Serveur\Codes\client.Py` »

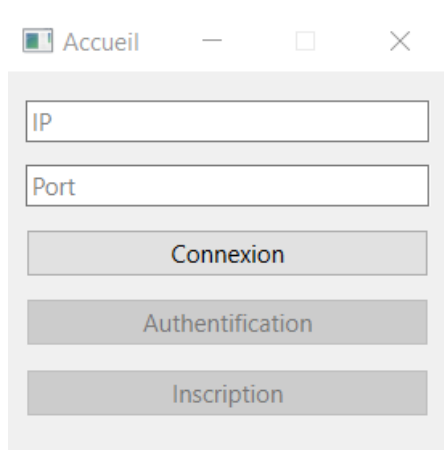
```
PS C:\Serveur\venv\Scripts> C:\Serveur\Codes\client.py
```

2. La deuxième, beaucoup plus simple, consiste à simplement récupérer et lancer le fichier exécutable « `client.exe` » du répertoire « `C:\Serveur\dist\` » comme suit :



Cet exécutable est une version compilée du programme client et contient à la fois un interpréteur Python adapté à nos besoins ainsi que toutes les dépendances requises à l'exécution. C'est beaucoup plus simple et pratique pour l'utilisateur.

Dans les deux cas, quelque soit la méthode d'exécution, un terminal client sera ouvert (qui affiche des prints), ainsi que la fenêtre d'accueil pour la connexion/authentification/inscription client :



C'est tout pour les procédures d'installation ! Il faudra maintenant vous référer aux procédures d'utilisations du serveur et du client.