

INSTALLATION SERVEUR (≈ 5 minutes)

STEP	Action	Commandes associées	Commentaire
1	Installer Python 3.10 minimum	<code>(root) -> apt-get install python3.10</code>	Il est impératif que la machine linux soit sur une version de python à minima supérieur à 3.10. Le script utilise une syntaxe qui n'est pas disponible dans les versions antérieures. Si une version antérieure à la 3.10 est utilisée, le programme renverra une erreur 'SyntaxError'
2	Vérifier si l'installation est complète	<code>(user) -> python3 --version</code>	Si la version affichée est inférieure à 3.10, vérifiez que la version installée a bien été ajoutée à la variable PATH du terminal. Redémarrer le terminal peut aider à la mise à jour de celle-ci. Sinon, vérifiez dans /etc/pythonX si la version 3.X est bien présente.
3	Installer / Mettre à jour le manager de bibliothèques 'pip'	<code>(root) -> apt-get install python3-pip</code>	N/A
4	Vérifier si l'installation est complète	<code>(user) -> pip3 --version</code>	Si la console renvoie un message d'erreur disant que pip n'existe pas, vérifiez que l'utilisateur est bien en ROOT lors de l'installation.
5	Installation de la bibliothèque 'Mysql-connector'	<code>(user) -> pip3 install mysql-connector-python</code>	N/A
6	Installation de Mariadb pour la gestion de la base de données	<code>(root) -> apt update</code> <code>(root) -> apt install mariadb-client mariadb-server</code>	N/A
7	Vérifier l'installation de mariadb	<code>(root) -> systemctl status mariadb</code>	Si le service est masqué : <code>(root) -> systemctl unmask mariadb</code> <code>(root) -> systemctl restart mariadb</code> <code>(root) -> systemctl status mariadb</code> Si le service est 'running', tout est bon. Si le service est 'stopped': <code>(root) -> systemctl start mariadb</code> <code>(root) -> systemctl status mariadb</code> En cas d'erreur, vérifiez le log à /var/mariadb/error.log pour déterminer la cause du problème.
8	Créer la base de données.	<code>(root) -> mariadb -u root -p</code> <code>(root) -> Appuyez sur 'Entrer'</code> <code>(Mariadb[None]) -> CREATE DATABASE SAE302;</code>	N/A
9	Créer l'utilisateur que nous utiliserons pour interagir avec la base de données.	<code>(Mariadb[None]) -> CREATE USER 'admin'@'localhost' IDENTIFIED BY 'admin'</code>	N/A

10	Donnez les droits à la bdd au nouvel utilisateur	<code>(Mariadb[None]) -> GRANT ALL PRIVILEGES ON SAE302.* TO 'admin'@'localhost' IDENTIFIED BY 'admin';</code>	Si une erreur de syntaxe est levée par mariadb, réessayez sans la partie "IDENTIFIED BY 'admin'"
11	Récupérez le template de base de données 'database_template.sql' sur le repo Github. Insérez le template de bdd sur Mariadb.	<code>(root) -> mariadb -u root -p SAE302 < database_template.sql</code>	N/A
12	Récupérez les fichiers du serveur sur le repo github (Dossier 'SERVER/')	Exemple en utilisant scp au cas où la machine hôte n'a pas d'interface graphique : <code>(root) -> scp -r SERVER/ <VM_user>@<VM_IP>:/home/<VM_user></code>	N/A
13	Entrez l'adresse IP du serveur.	Dans le fichier 'main.py' du dossier 'SERVER/' Allez à la ligne No 97, dans la fonction 'setup()'. A la ligne "server = Server_handle('0.0.0.0', 11111)" Remplacez 0.0.0.0 par l'adresse IP de la machine sur laquelle le serveur est installé. Il est très fortement conseillé de rendre cette adresse IP statique sur la machine hôte afin d'éviter que celle-ci change lors d'un renouvellement de bail DHCP par exemple. Il est également possible d'associer directement une adresse statique au serveur depuis le routeur dans les paramètres DHCP.	Il aurait été possible pour cette étape de directement prendre l'adresse locale du serveur via le script du serveur, cependant ceci peut mener à des résultats inconsistants suivant le type de machine. Par exemple, certaines VM prendront l'adresse de la machine hôte et non l'adresse bridgée de la VM ou bien prendront à la place l'adresse de loopback ce qui prévient les machines clients de rejoindre le serveur par la suite. Cette technique ne prend que quelques secondes de plus mais permet pour sûr d'associer la bonne adresse à la bonne machine. Il est également possible à l'avenir d'implémenter un système de broadcast UDP de l'adresse du serveur que celui-ci enverrait toutes les X secondes pour permettre aux clients d'écouter et de récupérer l'adresse avant de se connecter.
14	Démarrez le serveur !	<code>(user) -> cd ~/SERVER (user) -> python3 main.py</code>	Le serveur devrait maintenant être prêt et en train de tourner ! Vous pouvez également vous connecter sur la console en tant qu'admin avec les crédeniels 'admin'/'admin' pour exécuter des commandes. N.B Il n'est pas impératif de se connecter sur la console pour que les clients puissent interagir avec le serveur.