



Déploiement d'un serveur NodeJS avec BDD

Dans cette documentation nous allons voir comment mettre en place un serveur NodeJS avec une base donnée sur Ubuntu.

- I. [Installation des dépendances](#)
- II. [Installation du serveur et de la base de données](#)
- III. [Configuration](#)
- IV. [Mise en place du service](#)

Installation des dépendances

Pour commencer nous devons mettre à jour l'OS:

```
william@william-VirtualBox:~$ sudo apt update && sudo apt upgrade
```

Ensuite il faut installer les outils nécessaires :

```
william@william-VirtualBox:~$ sudo apt install nodejs npm mariadb-server git
```

Après ça il faut configurer l'installation de MySQL

```
william@william-VirtualBox:~$ sudo mysql_secure_installation

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user. If you've just installed MariaDB, and
haven't set the root password yet, you should just press enter here.

Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password or using the unix_socket ensures that nobody
can log into the MariaDB root user without the proper authorisation.

You already have your root account protected, so you can safely answer 'n'.

Switch to unix_socket authentication [Y/n] n
... skipping.

You already have your root account protected, so you can safely answer 'n'.

Change the root password? [Y/n] y
New password:
Re-enter new password:
Password updated successfully!
Reloading privilege tables..
... Success!

By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB without having to have a user account created for
them. This is intended only for testing, and to make the installation
go a bit smoother. You should remove them before moving into a
production environment.

Remove anonymous users? [Y/n]
... Success!

Normally, root should only be allowed to connect from 'localhost'. This
prevents that someone can guess at the root password from the network.
```

Installation du serveur et de la base donnée

En premier il faut télécharger le code du serveur déjà créer :

```
william@william-VirtualBox:~$ cd /opt/  
william@william-VirtualBox:/opt$ git clone https://github.com/Anas7823/m2l-back-end.git  
fatal: could not create work tree dir 'm2l-back-end': Permission denied  
william@william-VirtualBox:/opt$ sudo git clone https://github.com/Anas7823/m2l-back-end.git  
[sudo] password for william:
```

Puis installer les modules grâce à npm:

```
william@william-VirtualBox:/opt/m2l-back-end$ sudo npm install  
  
added 68 packages, and audited 69 packages in 8s  
  
7 packages are looking for funding  
run `npm fund` for details
```

Ensuite pour la BDD, il faut créer un utilisateur MariaDB ainsi que lui attribuer des privilèges afin de lui permettre l'accès à la base de données :

```
MariaDB [(none)]> CREATE USER 'william'@'localhost' IDENTIFIED BY 'Respons11';  
Query OK, 0 rows affected (0,024 sec)  
  
MariaDB [(none)]> GRANT ALL PRIVILEGES ON * . * TO 'william'@'localhost';  
Query OK, 0 rows affected (0,003 sec)  
  
MariaDB [(none)]> FLUSH PRIVILEGES;  
Query OK, 0 rows affected (0,007 sec)
```

Pour finir, il faut créer la base de données avant de pouvoir l'importer grâce au fichier SQL :

```
MariaDB [(none)]> CREATE DATABASE m2l;  
Query OK, 1 row affected (0,004 sec)  
  
MariaDB [(none)]> SHOW DATABASES  
+-----+  
| Database |  
+-----+  
| information_schema |  
| m2l |  
| mysql |  
| performance_schema |  
| sys |  
+-----+  
5 rows in set (0,023 sec)  
  
MariaDB [(none)]> Ctrl-C -- exit!  
Aborted  
william@william-VirtualBox:/opt$ mysql -u root -p m2l < /opt/m2l-back-end/m2l.sql  
Enter password:  
ERROR 1698 (28000): Access denied for user 'root'@'localhost'  
william@william-VirtualBox:/opt$ sudo mysql -u root -p m2l < /opt/m2l-back-end/m2l.sql
```

Configuration

Pour pouvoir se connecter à la BDD il nous faut un fichier `.env` contenant l'adresse de l'hôte de la BDD, le nom de la base, l'utilisateur et le mot de passe :

```
william@william-VirtualBox:/opt$ sudo touch .env
```

 (Créer le fichier)

```
sudo nano .env
```

 (Permet de modifier le fichier)

Ensuite, pour que le serveur se lance en même temps que la machine virtuelle nous allons créer un fichier service :

```
william@william-VirtualBox:/$ cd etc/  
william@william-VirtualBox:/etc$ cd systemd/system/  
william@william-VirtualBox:/etc/systemd/system$ sudo touch serveurnode.service
```

```
william@william-VirtualBox:/etc/systemd/system$ sudo nano serveurnode.service
```

Mise en place du service

Premièrement, nous allons utiliser `daemon-reload` afin de recharger les services :

```
william@william-VirtualBox:~$ sudo systemctl daemon-reload
```

Il faut ensuite activer le service

```
william@william-VirtualBox:~$ sudo systemctl enable serveurnode.service
```

Et le démarré :

```
william@william-VirtualBox:~$ sudo systemctl start serveurnode.service
```

Vous pouvez aussi vérifier son état grâce à la commande suivante :

```

william@william-VirtualBox:~$ sudo systemctl status serveurNode.service
[sudo] password for william:
● serveurNode.service - Serveur Node.js m2l
   Loaded: loaded (/etc/systemd/system/serveurNode.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2023-02-14 15:02:24 CET; 3min 34s ago
     Main PID: 564 (node)
        Tasks: 11 (limit: 2288)
      Memory: 64.7M
         CPU: 977ms
    CGroup: /system.slice/serveurNode.service
            └─564 /usr/bin/node /opt/m2l-back-end/Server.js

févr. 14 15:02:55 william-VirtualBox node[564]:      ColumnDef {
févr. 14 15:02:55 william-VirtualBox node[564]:        collation: [Collation],
févr. 14 15:02:55 william-VirtualBox node[564]:        columnLength: 11,
févr. 14 15:02:55 william-VirtualBox node[564]:        columnType: 3,
févr. 14 15:02:55 william-VirtualBox node[564]:        flags: 20489,
févr. 14 15:02:55 william-VirtualBox node[564]:        scale: 0,
févr. 14 15:02:55 william-VirtualBox node[564]:        type: 'INT'
févr. 14 15:02:55 william-VirtualBox node[564]:      }
févr. 14 15:02:55 william-VirtualBox node[564]:    ]
févr. 14 15:02:55 william-VirtualBox node[564]:  ]
lines 1-20/20 (END)...skipping...
● serveurNode.service - Serveur Node.js m2l
   Loaded: loaded (/etc/systemd/system/serveurNode.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2023-02-14 15:02:24 CET; 3min 34s ago
     Main PID: 564 (node)
        Tasks: 11 (limit: 2288)
      Memory: 64.7M
         CPU: 977ms
    CGroup: /system.slice/serveurNode.service
            └─564 /usr/bin/node /opt/m2l-back-end/Server.js

févr. 14 15:02:55 william-VirtualBox node[564]:      ColumnDef {
févr. 14 15:02:55 william-VirtualBox node[564]:        collation: [Collation],
févr. 14 15:02:55 william-VirtualBox node[564]:        columnLength: 11,
févr. 14 15:02:55 william-VirtualBox node[564]:        columnType: 3,
févr. 14 15:02:55 william-VirtualBox node[564]:        flags: 20489,
févr. 14 15:02:55 william-VirtualBox node[564]:        scale: 0,
févr. 14 15:02:55 william-VirtualBox node[564]:        type: 'INT'
févr. 14 15:02:55 william-VirtualBox node[564]:      }
févr. 14 15:02:55 william-VirtualBox node[564]:    ]
févr. 14 15:02:55 william-VirtualBox node[564]:  ]
févr. 14 15:02:55 william-VirtualBox node[564]:  ]

```