

# Gestion des sauvegardes

- [1. Création d'un serveur virtuel linux debian 11](#)
- [2. Installation de mariadb](#)
- [3. Création d'une base donnée en ligne de commande](#)
- [4. Export de la base de donnée et création d'un script bash](#)
- [5. Création d'un cron.](#)

## 1. Création d'un serveur virtuel linux debian 11

Pour créer un serveur virtuel il faut aller dans le logiciel **VM Virtualbox** de oracle.  
**Créer une VM** et y **ajouter la version debian 11** à cette machine virtuelle.

Puis il reste à faire la configuration du debian classique en y apportant les noms et mot de passe demandés.

## 2. Installation de mariadb

Une fois les configurations fini il nous faut un système de **Gestion de base de donnée** on a donc choisi de prendre MariaDB.

Pour l'installer nous nous sommes mis en super utilisateur (**sudo**) et nous avons utiliser la commande : **apt install mariadb-server**

```
on d'un utilisateur.  
root@mariadb:/home/mariadb# apt install mariadb-server  
lecture des listes de paquets... Fait  
Construction de l'arbre des dépendances... Fait  
lecture des informations d'état... Fait
```

### 3. Création d'une base donnée en ligne de commande

Une fois l'installation finie, nous pouvons nous connecter à mariadb et créer notre première base de données appelé sauv.

commande : **create database sauv;**

```
root@mariadb:/home/mariadb# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 30
Server version: 10.5.15-MariaDB-0+deb11u1 Debian 11

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> create database sauv
-> ;
Query OK, 1 row affected (0.000 sec)

MariaDB [(none)]> _
```

Pour vérifier son existence on peut faire la commande **show databases.**

```
MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sauv |
+-----+
4 rows in set (0.000 sec)
```

Pour l'exemple nous avons inséré quelques données dans une table nommé **“etat”** dans la bdd **sauv.**

```
MariaDB [sauv]> alter table etat add primary key (id);
Query OK, 0 rows affected (0.109 sec)
Records: 0 Duplicates: 0 Warnings: 0

MariaDB [sauv]> insert into etat values('CL', 'fiche frais'), ('RE', 'refuser');
Query OK, 2 rows affected (0.013 sec)
Records: 2 Duplicates: 0 Warnings: 0

MariaDB [sauv]> _
```

## 4. Export de la base de donnée et création d'un script bash

Une fois que notre base de donnée est rempli nous pouvons faire l'export dans un fichier nommé `dump_filename.sql`.

avec la commande : **`mysqldump -u root -p sauv > dump_filename.sql`**

```
root@mariadb:/home/mariadb# mysqldump -u root -p sauv > dump_filename.sql
Enter password:
root@mariadb:/home/mariadb# ls
dump_filename.sql
root@mariadb:/home/mariadb# _
```

la commande : **`ls`** permet de vérifier de sa création

On peut maintenant passer à la création d'un script bash (.sh) pour faire cela de manière automatique.

```
root@GestBDD:/# chmod +x usr/local/bin/mysqldump.sh
```

Le fichier **`mysqldump.sh`** permettra de **sauvegarder la base de données** d'un dossier appelé "**`mariadb`**". Ici nous lui donnons les droits d'exécution pour pouvoir l'exécuter.

```
GNU nano 5.4 /usr/local/bin/mysqldump.sh
#!/bin/bash
mysqldump -u astik -p root sauv > /var/backups/mariadb/$(date +%Y%m%d-%H%M%S)-backup.sql
```

## 5. Création d'un cron.

Pour le faire de manière automatique maintenant nous pouvons utiliser le fichier crontab qui permet de faire l'exécution d'un script pour les temps x temps défini dans la commande.

Ici nous faisons la procédure toutes les **2heures**.

```
GNU nano 5.4 /tmp/crontab.7cnMtp/crontab *
#TOUS LES JOURS à 2 HEURES DU MATIN
0 2 * * * /usr/local/bin/mysqldump.sh >/dev/null 2>&1
```

Il exécute le fichier bash.