TP - Backup MariaDB

Objectif :

- Apprendre à coder un script Bash pour exécuter des commandes Linux
- Sécuriser MariaDB
- Faire une sauvegarde des données

Prérequis :

- Connaitre les commandes linux

Étapes du TP :

1. Vous devez créer une machine virtuelle sous linux Debian avec les informations suivantes :

> ⓘ **Téléchargement de Debian 13**
>
> https://www.debian.org/download

Informations	Valeur
nom de la machine	mariadb-serv-01
Domain	lerebours.lan
RAM	2Go
Disque	50Go
CPU	1
Core	1
root	Nk75LeR!
user	maria
pwd	Nk44LeR!

2. Mettre le user "maria" dans l'autorisation "sudo"
3. Installation de la base de donnée MariaDB

Mettez à jour vos paquets système :
Il est important de s'assurer que votre système est à jour avant d'installer de nouveaux logiciels.

```
    sudo apt update -y && sudo apt upgrade -y
```

Installer le paquet serveur MariaDB :
Les dépôts officiels de Debian 13 incluent le paquet `mariadb-server`, vous pouvez donc l'installer directement.

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

Démarrer et activer le service MariaDB :
Ceci garantit que MariaDB s'exécute automatiquement au démarrage de votre système.

```
sudo systemctl start mariadb
sudo systemctl enable mariadb
```

Sécuriser l'installation de MariaDB :
Exécutez le script de sécurité pour définir le mot de passe root, supprimer les utilisateurs anonymes, interdire la connexion root à distance, supprimer la base de données de test et recharger les tables de privilèges. Suivez les instructions pour renforcer la sécurité.

```
sudo mariadb-secure-installation
```

Informations	Valeur
Enter root user password or leave blank :	Nk75LeR!
Change the root password ?	Db44LeR?

Les étapes :

```
NOTE: MariaDB is secure by default in Debian. Running this script
is
      useless at best, and misleading at worst. This script will
```

```
be
      removed in a future MariaDB release in Debian. Please read
      mariadb-server.README.Debian for details.

Enter root user password or leave blank:

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]
Enabled successfully (or at least no errors was emitted)!
Reloading privilege tables..
 ... Success!


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

Change the root password? [Y/n]
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]
SQL executed without errors!
The operation might have been successful, or it might have not
done anything.

Normally, root should only be allowed to connect from
'localhost'.  This
ensures that someone cannot guess at the root password from the
network.

Disallow root login remotely? [Y/n]
SQL executed without errors!
The operation might have been successful, or it might have not
done anything.

By default, MariaDB comes with a database named 'test' that
anyone can
access.  This is also intended only for testing, and should be
removed
before moving into a production environment.

Remove test database and access to it? [Y/n]
 - Dropping test database...
SQL executed without errors!
The operation might have been successful, or it might have not
done anything.
 - Removing privileges on test database...
SQL executed without errors!
The operation might have been successful, or it might have not
done anything.

Reloading the privilege tables will ensure that all changes made
so far
```

```
will take effect immediately.

Reload privilege tables now? [Y/n]
 ... Success!

Cleaning up...

All done!  If you've completed all of the above steps, your
MariaDB
installation should now be secure.

Thanks for using MariaDB!
```

Vérifier l'état du service MariaDB :
Vérifiez si le service MariaDB fonctionne correctement.

```
sudo systemctl status mariadb
```

(Facultatif) Créer une base de données et un utilisateur :
Si vous avez besoin de créer une nouvelle base de données et un nouvel utilisateur, vous pouvez le faire en vous connectant à MariaDB et en exécutant des commandes SQL.
Se connecter à MariaDB :

```
mysql -u root -p
```

Informations	Valeur
user	gestroot
mot de passe	Db8844LeR?
Nom database	gest_travaux

Puis exécutez les commandes SQL :

```
CREATE DATABASE example_database;
CREATE USER 'example_user'@'localhost' IDENTIFIED BY
'StrongPasswordUserHere';
```

```
GRANT ALL ON example_database.* TO 'example_user'@'localhost'
WITH GRANT OPTION;
FLUSH PRIVILEGES;
EXIT;
```

Vous pouvez ajuster la configuration de MariaDB dans `/etc/mysql/mariadb.conf.d/50-server.cnf` si nécessaire, puis redémarrer le service avec `sudo systemctl restart mariadb`.

Sources

Annexes

Passer en root

```
su -
```

Listez les bases de données dans MariaDB :

```
SHOW DATABASES;
```

Listez les utilisateurs dans MariaDB :

```
SELECT User, Host FROM mysql.user;
```

Affichez les droits de l'utilisateur courant dans MariaDB :

```
SHOW GRANTS FOR CURRENT_USER;
```

Affichez les droits d'un utilisateur spécifique dans MariaDB

```
SHOW GRANTS FOR 'example_user'@'*';
```

Connexions externes au serveur MariaDB

Pour autoriser les connexions externes au serveur MariaDB, vous devez modifier son fichier de configuration pour qu'il écoute sur toutes les interfaces réseau disponibles, et non plus seulement sur `localhost`.

Voici les étapes à suivre :

1. **Connectez-vous à votre serveur MariaDB :**
   Utilisez SSH pour vous connecter à votre serveur Debian 13.
2. **Modifiez le fichier de configuration de MariaDB :**
   Le fichier de configuration principal pour le serveur MariaDB se trouve généralement dans `/etc/mysql/mariadb.conf.d/50-server.cnf`. Vous devrez l'éditer avec un éditeur de texte comme `nano` ou `vim`.

   ```
   sudo nano /etc/mysql/mariadb.conf.d/50-server.cnf
   ```
3. **Localisez et modifiez la directive `bind-address` :**
   Dans ce fichier, recherchez la ligne qui commence par `bind-address`. Par défaut, elle est souvent définie comme suit pour restreindre l'accès à `localhost` :

   ```
   bind-address = 127.0.0.1
   ```

   Pour autoriser les connexions depuis l'extérieur, vous avez deux options principales :
   - **Pour autoriser les connexions depuis n'importe quelle adresse IP** (ce qui est moins sécurisé et doit être utilisé avec prudence, en combinant avec un pare-feu) :
     Changez la ligne en :
     ```
     bind-address = 0.0.0.0
     ```
   - **Pour autoriser les connexions depuis une adresse IP spécifique** (plus sécurisé) :
     Remplacez `127.0.0.1` par l'adresse IP publique ou privée de votre serveur sur laquelle vous souhaitez que MariaDB écoute. Par exemple :
     ```
     bind-address = VOTRE_ADRESSE_IP_EXTERNE
     ```
     Si vous n'êtes pas sûr de l'adresse IP à utiliser, `0.0.0.0` est souvent utilisé pour écouter sur toutes les interfaces disponibles.
4. **Sauvegardez et quittez le fichier :**
   Dans `nano`, appuyez sur `Ctrl+X`, puis `Y` pour confirmer la sauvegarde, et `Entrée` pour enregistrer le nom du fichier.
5. **Redémarrez le service MariaDB :**
   Pour que les modifications prennent effet, vous devez redémarrer le service MariaDB :

```
sudo systemctl restart mariadb
```

6. **(si nécessaire) Configurez le pare-feu :**
   Assurez-vous que votre pare-feu (comme `ufw` ou `iptables`) autorise les connexions entrantes sur le port de MariaDB, qui est généralement le port `3306`. Par exemple, avec `ufw` :

```
sudo ufw allow 3306/tcp
sudo ufw reload
```

7. **Créez des utilisateurs avec des droits d'accès appropriés :**
   Il est fortement recommandé de ne pas utiliser l'utilisateur `root` pour les connexions externes. Créez des utilisateurs spécifiques avec les privilèges nécessaires pour les applications qui doivent accéder à la base de données depuis l'extérieur. Par exemple :

```
-- Connectez-vous à MariaDB
mysql -u root -p

-- Créez un nouvel utilisateur qui peut se connecter depuis
n'importe où ('%') ou une IP spécifique
CREATE USER 'remote_user'@'%' IDENTIFIED BY
'votre_mot_de_passe_securise';
-- Ou pour une IP spécifique :
-- CREATE USER 'remote_user'@'192.168.1.100' IDENTIFIED BY
'votre_mot_de_passe_securise';

-- Accordez les privilèges nécessaires (par exemple, tous les
privilèges sur une base de données spécifique)
GRANT ALL PRIVILEGES ON votre_base_de_donnees.* TO
'remote_user'@'%';
-- Ou pour une IP spécifique :
-- GRANT ALL PRIVILEGES ON votre_base_de_donnees.* TO
'remote_user'@'192.168.1.100';

-- Appliquez les changements
FLUSH PRIVILEGES;
EXIT;
```

Remplacez `'remote_user'`, `'%'` (ou l'adresse IP spécifique), `'votre_mot_de_passe_securise'` et `'votre_base_de_donnees'` par vos propres valeurs. L'utilisation de `'%'` comme hôte permet la connexion depuis n'importe quelle adresse IP, ce qui est pratique mais moins sécurisé. Il est préférable de spécifier une adresse IP ou une plage d'adresses IP si possible.

En suivant ces étapes, vous devriez pouvoir accéder à votre serveur MariaDB depuis des machines externes.