
Documentation Technique - Projet de Sauvegarde MariaDB

Table des matières

- Objectifs du projet
 - Architecture du système
 - Étapes de réalisation
 - Étape 1 : Installation de MariaDB (VM1)
 - Étape 2 : Création de la base de données (VM1)
 - Étape 3 : Création d'un utilisateur pour la DB (VM1)
 - Étape 4 : Insertion d'un jeu de données (VM1)
 - Étape 5 : Configuration SSH entre les VMs
 - Étape 6 : Création du script Bash de sauvegarde
 - Étape 7 : Test du script de sauvegarde
 - Conclusion
-

Objectifs du projet

Objectifs :

- Installer et gérer un serveur de base de données MariaDB
 - Rédiger un script de sauvegarde en bash
 - Utiliser SSH et SCP pour le transfert sécurisé des sauvegardes
-

Architecture du système

Description de l'infrastructure

💡 VM 1 (Serveur de base de données) :

- Une base de données MariaDB
- Un jeu de données de test
- Un script bash de sauvegarde

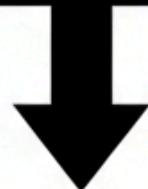
💡 VM 2 (Serveur de stockage) :

- Serveur SSH configuré
- Réception des sauvegardes depuis VM1

Schéma de communication

VM 2 (Stockage)

- OpenSSH Server
- Réception des dumps
- Stockage des sauvegardes



SSH (connexion depuis VM2)

VM 1 (Base de données)

- MariaDB Server
- Base de données + données
Script backup.sh
- Répertoire /srv/backups



VM 2 (destination) SCP (envoi dump)

Flux de sauvegarde

1. **Dump de la base** : Extraction des données avec `mysqldump`
2. **Stockage local** : Placement dans `/srv/backups/` sur VM1
3. **Renommage** : Format `dump_AAAA_MM_JJ.sql` (ex: `dump_1980_01_01.sql`)
4. **Transfert SCP** : Envoi du dump vers VM2

Étapes de réalisation

Étape 1 : Installation de MariaDB (VM1)

La première étape consiste à installer le serveur de base de données MariaDB sur la VM1.

1.1 Mise à jour du système et installation

Sur la **VM1**, commencez par mettre à jour les paquets puis installer MariaDB Server.

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

```
medusa@0xarrogant:~$ sudo apt update && sudo apt install mariadb-server -y
```

1.2 Vérification du service MariaDB

Vérifiez que le service MariaDB est actif et fonctionnel.

```
sudo systemctl status mariadb
```

```
medusa@0xarrogant:~$ sudo systemctl status mariadb
[sudo] Mot de passe de medusa :
● mariadb.service - MariaDB 10.11.14 database server
  Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; preset:
  Active: active (running) since Thu 2026-01-29 10:15:52 CET; 3h 59min ago
    Docs: man:mariadb(8)
          https://mariadb.com/kb/en/library/systemd/
   Main PID: 1764 (mariadb)
     Status: "Taking your SQL requests now..."
      Tasks: 9 (limit: 30003)
     Memory: 145.9M (peak: 153.5M)
        CPU: 4.239s
       CGroup: /system.slice/mariadb.service
                 └─1764 /usr/sbin/mariadb

janv. 29 10:15:52 0xarrogant mariadb[1764]: 2026-01-29 10:15:52 0 [Note] Plug
janv. 29 10:15:52 0xarrogant mariadb[1764]: 2026-01-29 10:15:52 0 [Warning] Y
janv. 29 10:15:52 0xarrogant mariadb[1764]: 2026-01-29 10:15:52 0 [Note] Serv
janv. 29 10:15:52 0xarrogant mariadb[1764]: 2026-01-29 10:15:52 0 [Note] /usr
janv. 29 10:15:52 0xarrogant mariadb[1764]: Version: '10.11.14-MariaDB-0ubunt
```

Étape 2 : Création de la base de données (VM1)

Une fois MariaDB installé, nous allons créer une base de données pour notre projet.

2.1 Connexion à MariaDB

Connectez-vous au serveur MariaDB en tant qu'utilisateur root.

```
sudo mysql -u root
```

```
medusa@0xarrogant:~$ sudo mysql -u root
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 33
Server version: 10.11.14-MariaDB-0ubuntu0.24.04.1 Ubuntu 24.04

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)]> █
```

2.2 Création de la base de données

Créez une nouvelle base de données nommée `projet_db`.

```
CREATE DATABASE projet_db;
```

2.3 Vérification de la création

Affichez la liste des bases de données pour confirmer.

```
SHOW DATABASES;
```

```
MariaDB [(none)]> SHOW DATABASES;
```

Database
bts_project
bts_sio_db
glpi
information_schema
mysql
performance_schema
projet_db
sys
testdata

```
9 rows in set (0,034 sec)
```

```
MariaDB [(none)]> █
```

Étape 3 : Création d'un utilisateur pour la DB (VM1)

Pour des raisons de sécurité, nous créons un utilisateur dédié pour gérer la base de données.

3.1 Crédit de l'utilisateur

Créez un utilisateur SQL avec un mot de passe sécurisé.

```
CREATE USER 'stan'@'localhost' IDENTIFIED BY 'T!G#!M';
```

[IMAGE : Confirmation de création de l'utilisateur]

3.2 Attribution des privilèges

Donnez tous les privilèges sur `projet_db` à l'utilisateur `admin_user`.

```
GRANT ALL PRIVILEGES ON projet_db.* TO 'stan'@'localhost';
FLUSH PRIVILEGES;
```

[IMAGE : Confirmation d'attribution des privilèges]

3.3 Vérification des privilèges

Vérifiez que les privilèges ont bien été attribués.

```
SHOW GRANTS FOR 'stan'@'localhost';
```

```
MariaDB [(none)]> SHOW GRANTS FOR 'stan'@'localhost';
+-----+
| Grants for stan@localhost
|
+-----+
| GRANT USAGE ON *.* TO `stan`@`localhost` IDENTIFIED BY PASSWORD '*B1A201B0B343
961A4DF50D46E3E9192F61F31F31' |
| GRANT ALL PRIVILEGES ON `projet_db`.* TO `stan`@`localhost` |
+-----+
-----+
2 rows in set (0,022 sec)
```

Étape 4 : Insertion d'un jeu de données (VM1)

Maintenant, créons une table et insérons des données de test.

4.1 Sélection de la base de données

```
USE projet_db;
```

```
MariaDB [(none)]> USE projet_db;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A
```

```
Database changed
```

```
MariaDB [projet_db]>
```

4.2 Création d'une table

Créez une table `utilisateurs` avec plusieurs champs.

```
CREATE TABLE utilisateurs (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nom VARCHAR(50) NOT NULL,
    prenom VARCHAR(50) NOT NULL,
    email VARCHAR(100) UNIQUE
);
```

[IMAGE : Confirmation de création de la table]

4.3 Insertion de données de test

Insérez au moins deux enregistrements dans la table.

```
INSERT INTO utilisateurs (nom, prenom, email) VALUES
('Dupont', 'Alice', 'alice.dupont@example.fr'),
('Martin', 'Bob', 'bob.martin@example.fr');
```

[IMAGE : Confirmation d'insertion : Query OK, 2 rows affected]

4.4 Vérification des données

Affichez les données insérées pour vérifier.

```
SELECT * FROM utilisateurs;
```

```
MariaDB [(none)]> SELECT * FROM utilisateurs;
ERROR 1046 (3D000): No database selected
MariaDB [(none)]>
MariaDB [(none)]> USE projet_db;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A
```

Database changed

```
MariaDB [projet_db]> SELECT * FROM utilisateurs;
+----+-----+-----+
| id | nom   | email          |
+----+-----+-----+
| 1  | Alice | alice@example.com |
| 2  | Bob   | bob@example.com  |
+----+-----+-----+
2 rows in set (0,016 sec)
```

```
MariaDB [projet_db]>
```

4.5 Sortie de MariaDB

```
EXIT;
```

Étape 5 : Configuration SSH entre les VMs

Pour permettre le transfert automatique des sauvegardes, nous devons configurer SSH.

5.1 Installation d'OpenSSH Server sur VM2

Sur la **VM2**, installez le serveur SSH si ce n'est pas déjà fait.

```
sudo apt update  
sudo apt install openssh-server -y
```

```
medusa@0xarrogant:~$ sudo apt update && sudo apt install openssh-server -y  
Réception de :1 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]  
Atteint :2 http://archive.ubuntu.com/ubuntu noble InRelease  
Réception de :3 http://archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]  
]  
Réception de :4 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [1 410 kB]
```

5.2 Vérification du service SSH sur VM2

```
sudo systemctl status ssh
```

```
medusa@0xarrogant:~$ sudo systemctl status ssh
[sudo] Mot de passe de medusa :
Warning: The unit file, source configuration file or drop-ins of ssh.service change
● ssh.service - OpenBSD Secure Shell server
    Loaded: loaded (/usr/lib/systemd/system/ssh.service; disabled; preset: enable)
      Active: inactive (dead)
TriggeredBy: ● ssh.socket
    Docs: man:sshd(8)
           man:sshd_config(5)
lines 1-7/7 (END)
```

5.3 Connexion de VM2 vers VM1 (test)

Depuis la **VM2**, testez la connexion SSH vers VM1 pour vérifier la connectivité.

```
ssh medusa@192.168.20.129
```

```
medusa@medusa-VMware-Virtual-Platform:~/backups$ ssh medusa@192.168.20.129
The authenticity of host '192.168.20.129 (192.168.20.129)' can't be established.
ED25519 key fingerprint is SHA256:lRvboiNlguFq9KAJ8ls+Pag4co3iSFAyhIav03vpkmE.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? s
```

Tapez `yes` puis entrez le mot de passe. Si la connexion réussit, tapez `exit` pour revenir sur VM2.

5.4 Génération de clés SSH sur VM1

Sur la **VM1**, générez une paire de clés SSH pour permettre l'authentification sans mot de passe.

```
ssh-keygen -t rsa -b 4096
```

Appuyez sur **Entrée** pour accepter l'emplacement par défaut et laissez la passphrase vide (appuyez deux fois sur Entrée).

5.5 Copie de la clé publique vers VM2

Copiez la clé publique de VM1 vers VM2 pour autoriser les connexions sans mot de passe.

```
ssh-copy-id medusa@192.168.20.128
```

Entrez le mot de passe de l'utilisateur sur VM2 lorsque demandé.

5.6 Test de connexion SSH sans mot de passe

Testez la connexion depuis VM1 vers VM2 sans saisir de mot de passe.

```
ssh medusa@192.168.20.128
```

Si tout est correct, vous devriez être connecté directement **sans demande de mot de passe**.

Tapez `exit` pour revenir sur VM1.

Étape 6 : Création du script Bash de sauvegarde

Le script bash doit effectuer les opérations suivantes :

1. Faire un dump de la base de données
2. Déplacer le dump vers `/srv/backups`
3. Renommer le fichier au format `dump_AAAA_MM_JJ.sql`
4. Envoyer le dump vers VM2 via SCP

6.1 Crédit du répertoire de sauvegarde

Sur la **VM1**, créez le répertoire `/srv/backups`.

```
sudo mkdir -p /srv/backups  
sudo chown $USER:$USER /srv/backups
```

6.2 Cration du script backup.sh

Crez le fichier script dans votre repertoire personnel.

```
nano ~/backup.sh
```

6.3 Contenu du script

Copiez le contenu suivant dans le fichier :

```
https://github.com/
```

[IMAGE : Contenu du fichier backup.sh dans l'diteur nano]

Appuyez sur `Ctrl + O` pour enregistrer, puis `Entree`, puis `Ctrl + X` pour quitter.

6.4 Rendre le script executable

Ajoutez les permissions d'execution au script.

```
chmod +x ~/backup.sh
```

tape 7 : Test du script de sauvegarde

7.1 Execution du script

Lancez le script de sauvegarde.

```
./backup.sh
```

Sortie attendue après avoir rempli toute les demandes de l'application GUI :



7.2 Vérification sur VM1

Vérifiez que le dump est bien présent dans `/srv/backups/`.

```
ls -lh /srv/backups/
```

```
medusa@0xarrogant:~$ ls -lh /srv/backups/
total 8,0K
-rw-rw-r-- 1 medusa medusa 2,1K janv. 29 15:10 dump_20260129_1510.sql
-rw-rw-r-- 1 medusa medusa 2,1K janv. 29 11:19 dump_2026_01_29.sql
-rw-rw-r-- 1 medusa medusa     0 janv. 22 17:34 dump_testdb_2026-01-22.sql
```

7.3 Vérification sur VM2

Connectez-vous à VM2 et vérifiez que le fichier a bien été reçu.

```
ssh utilisateur@IP_VM2  
ls -lh ~/
```

Conclusion

Objectifs atteints

Ce projet a permis de mettre en place une solution complète de sauvegarde automatisée pour une base de données MariaDB. Tous les objectifs ont été remplis :

1. Installation et gestion d'un serveur de base de données

- Installation de MariaDB sur VM1
- Création d'une base de données `projet_db`
- Création d'un utilisateur SQL dédié avec privilèges appropriés
- Insertion d'un jeu de données de test

2. Rédaction d'un script de sauvegarde en bash

- Script capable de faire un dump de la base de données
- Déplacement automatique vers `/srv/backups`
- Renommage au format `dump_AAAA_MM_JJ.sql`
- Transfert sécurisé via SCP vers VM2

3. Utilisation de SSH et SCP

- Configuration de l'authentification par clé SSH
- Transfert sécurisé des fichiers entre les deux VMs
- Automatisation sans intervention manuelle

Projet réalisé dans le cadre d'un TP Cyber