

Mise en place et configuration d'un serveur MariaDB (Mission 5)

Configuration du container

1. Contexte

L'organisation a besoin de deux serveurs de bases de données (MariaDB), avec des configurations précises : - Noms d'hôte : priv-db1 et priv-db2 - IP : 10.31.208.33 et 10.31.208.34 - Sécurité et gestion des accès par utilisateurs - Accès distant autorisé pour certains comptes

2. Configuration réseau et VM

Chaque VM a été créée dans Proxmox, avec : - Adresse IP fixe (exemple pour priv-db2) : 10.31.208.34/22 - Vérification avec `ip a` et `ping`

Configuration Maria db

3. Installation et sécurisation de MariaDB

Mise à jour de la machine :

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

Installation de MariaDB :

```
apt install mariadb-server mariadb-client -y
```

Vérification du service :

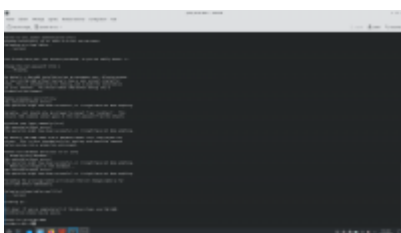
```
systemctl status mariadb
```

Démarrage :

```
systemctl start mariadb
```

sécurisation de l'installation

```
mariadb-secure-installation
```



Il faut juste appuyer sur entrée cela laissera les parametre par default.

4. Création de la base et des comptes

- Base de données : `CREATE DATABASE gsb;` - Compte admin : accès total et gestion des privilèges - Compte gsb : accès limité uniquement à la base gsb - Compte test : créé avec plugin ed25519 pour sécurité renforcée

Connexion à MariaDB en root On entre dans le shell MariaDB :

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

5. Création des utilisateurs a) Création de l'utilisateur administrateur

```
CREATE USER 'admin'@'%' IDENTIFIED BY 'drowssap';
```

Lui donner tous les privilèges (administration complète) :

```
GRANT ALL PRIVILEGES ON *.* TO 'admin'@'%' WITH GRANT OPTION;
```

b) Création de l'utilisateur applicatif GSB

```
CREATE USER 'gsb'@'%' IDENTIFIED BY 'drowssap';
```

(On n'a pas mis de privilèges précis, mais selon les besoins on pourrait restreindre à une seule base).

c) Suppression d'un ancien utilisateur (ex. test@localhost) Si jamais il existe encore :

```
DROP USER 'test'@'localhost';
```

d) Exemple de création d'un utilisateur avec un plugin (non retenu ici) On avait tenté :

```
CREATE USER 'test'@'localhost' IDENTIFIED VIA ed25519 USING  
PASSWORD('drowssap');
```

Cette commande avait été tapée directement dans le shell Linux, d'où l'erreur bash: syntax error. Il faut toujours exécuter ce genre de commandes depuis l'invite MariaDB [(none)]> et pas dans le terminal root Linux. Validation des privilèges

```
FLUSH PRIVILEGES;
```

Vérification des utilisateurs Lister les utilisateurs créés :

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



5. Configuration des accès distants

Par défaut, MariaDB écoute uniquement en local (127.0.0.1). Il faut modifier le fichier de configuration :

```
nano /etc/mysql/mariadb.conf.d/50-server.cnf
```

Chercher la ligne :

```
bind-address = 127.0.0.1
```

Et la remplacer par :

```
bind-address = 0.0.0.0
```

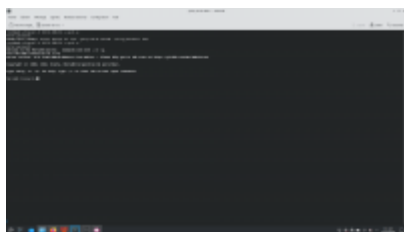
Ensuite, redémarrer MariaDB :

```
systemctl restart mariadb
```



6. Tests

Connexion avec gsb : accès uniquement à gsb



7. Pourquoi '%' plutôt que 'localhost'

- `user@localhost` : accès uniquement local au serveur
- `user@%` : accès depuis n'importe quelle IP (nécessaire ici)

8. Validation

- Création conteneur - Configuration IP - Installation MariaDB - Sécurisation initiale - Création comptes (admin, gsb, test) - Configuration accès distants

Conclusion La mission est remplie : les deux serveurs priv-db1 et priv-db2 sont opérationnels, les utilisateurs répondent au cahier des charges, et l'accès distant fonctionne.

Last
update:
2025/12/05 11:16 sistr2-oceanie:installation_et_configuration_de_mariadb https://sistr2.beaupeyrat.com/doku.php?id=sistr2-oceanie:installation_et_configuration_de_mariadb

From:
<https://sistr2.beaupeyrat.com/> - **Documentations SIO2 option SISR**

Permanent link:
https://sistr2.beaupeyrat.com/doku.php?id=sistr2-oceanie:installation_et_configuration_de_mariadb

Last update: **2025/12/05 11:16**

