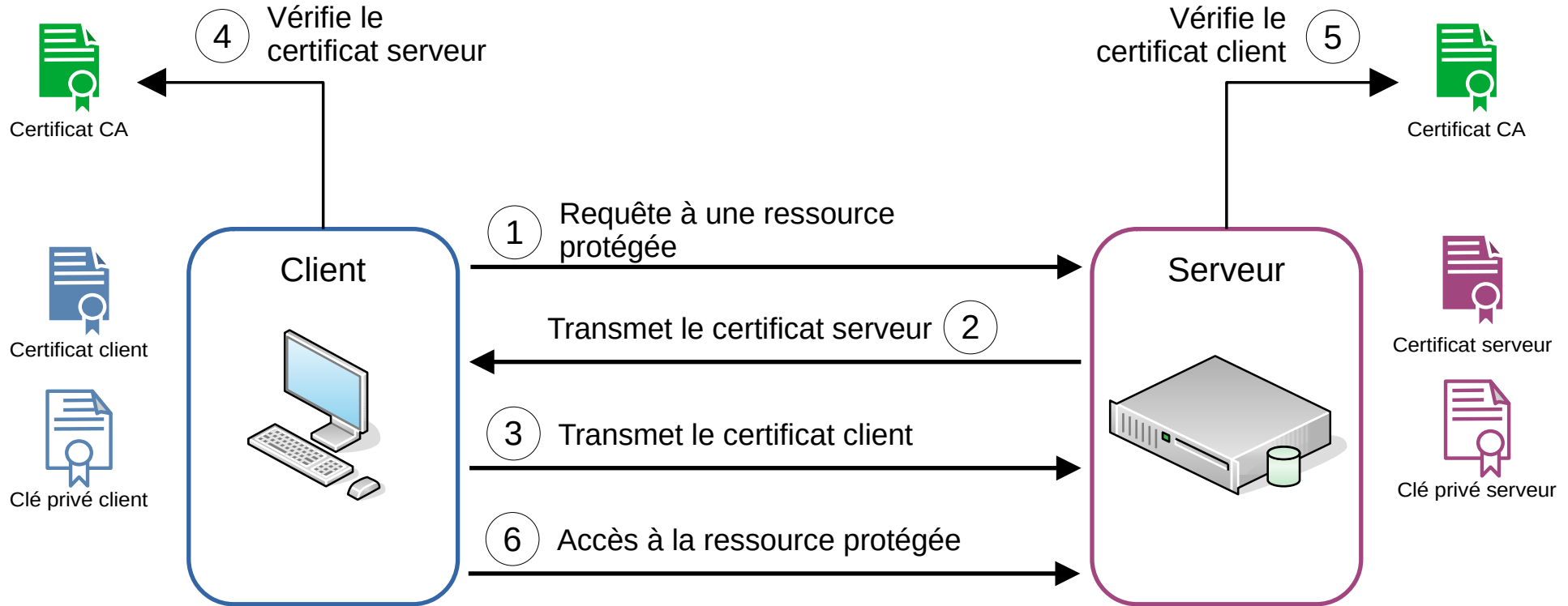


# Comment sécuriser une connexion distante sur Mariadb ?

# Que cherche t-on a obtenir ?

Le client et le serveur s'authentifie mutuellement via un certificat d'Autorité auto-signé.



# Check-list

Côté serveur :

**server-cert.pem**: certificat SSL du serveur

**server-key.pem**: clé privée du serveur

**ca-cert.pem**: autorité de certification (CA) qui a signé le certificat du serveur

Vérifier les droits d'accès à ces fichiers

Configurer le fichier `/etc/mysql/mariadb.conf.d/50-server.cnf`

Un utilisateur SQL configurer pour se connecter au serveur sécurisé

Un utilisateur système pour le transfert des clés via une connexion SSH

Côté client :

Récupérer sur le serveur le certificat, la clé client et le certificat CA via SSH

**client-cert.pem**: certificat SSL du client

**client-key.pem**: clé privée du client

**ca-cert.pem**: autorité de certification (CA) qui a signé le certificat du client

Vérifier les droits d'accès à ces fichiers

Configurer le fichier `~/.mysql/my.cnf` ou un client SQL

# Prérequis sur le serveur

## Vérifiez que le service mariadb.service soit actif

```
sudo systemctl status mariadb.service
```

```
● mariadb.service - MariaDB 10.11.2 database server
   Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; vendor preset: enabled)
   Drop-In: /etc/systemd/system/mariadb.service.d
            └─migrated-from-my.cnf-settings.conf
lines 1-4
```

## Vérifiez que le port réseau 3306 soit à l'écoute

```
sudo netstat -tulnp | grep 3306
```

```
Tcp      0      0 0.0.0.0:3306      0.0.0.0:*        LISTEN      27690/mariabdb
```

## Vérifiez que le serveur ping le client

```
ping <IP-DU-CLIENT>
```

## Activer le serveur SSH

```
sudo apt-get update
```

```
sudo apt-get install openssh-server
```

# Prérequis sur le serveur

## Créez un compte système pour le transfert de fichier

```
sudo adduser nom_utilisateur
```

## Configurez SSH

```
sudo nano /etc/ssh/sshd_config
```



```
...  
PermitRootLogin no  
...  
PasswordAuthentication yes  
...  
AllowUsers nom_utilisateur
```

## Nous avons :

- Interdit les connexions SSH avec le compte root
- Activé l'authentification des utilisateurs système
- Autorisé un nouvel utilisateur à se connecter via SSH sur cette machine

## Redémarrez SSH

```
sudo systemctl restart ssh
```

# Prérequis sur le client

**Vérifiez que le client ping le serveur**

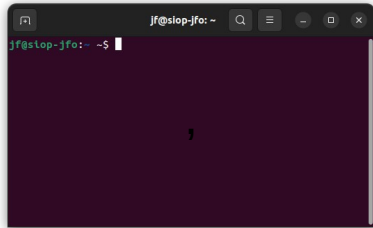
```
ping <IP-DU-SERVEUR>
```



**DBeaver**

<https://dbeaver.io>

**Assurez-vous qu'un « client SQL » soit installé sur le client**



```
sudo apt install mariadb-client
```



**Workbench**



<https://antares-sql.app/>

# Configuration du serveur Mariadb pour accepter les connexions distantes

➡ Depuis le client, testez la connexion au serveur Mariadb



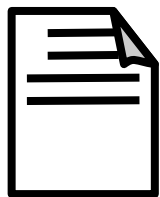
```
jf@siop-jfo:~ ~$ mysql -u root -p -h 172.16.254.151
Enter password:
ERROR 2002 (HY000): Can't connect to server on '172.16.254.151' (115)
```

**ERROR 2002 (HY000):** Can't connect to server on '172.16.254.151' (115)

<https://mariadb.com/kb/en/mariadb-error-codes/>

➡ Depuis le serveur, éditez le fichier 50-server.cnf

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



[mysqld]

...

→ **Bind-address = 0.0.0.0**

...

<https://mariadb.com/kb/en/configuring-mariadb-for-remote-client-access/>

➡ Redémarrez le serveur Mariadb

```
sudo systemctl restart mariadb.service
```

# Configuration du serveur Mariadb pour accepter les connexions distantes

➡ Depuis le client, tentez une connexion SQL au serveur Mariadb



```
jf@siop-jfo:~ ~$ mysql -u root -p -h 172.16.254.151
Enter password: 0000000000000000
ERROR 1130 (HY000): Host '10.200.6.11' is not allowed to connect to this MariaDB server
```

**ERROR 1130 (HY000):** Host '10.200.6.11' is not allowed to connect to this MariaDB server  
<https://mariadb.com/kb/en/mariadb-error-codes/>

➡ Depuis le serveur mariadb

1) Connectez vous au serveur SQL

2) Affichez la liste des utilisateurs

```
SELECT user,host FROM mysql.user;
```

```
MariaDB [(none)]> SELECT user,host FROM mysql.user;
+-----+-----+
| user | host      |
+-----+-----+
| root | localhost |
+-----+-----+
1 row in set (0.000 sec)
```

3) Ajoutez un utilisateur SQL autorisé à se connecter à distance

```
GRANT USAGE ON *.* TO 'user'@'%' IDENTIFIED BY 'mariadb';
```

➡ Depuis le client, tentez une connexion SQL au serveur Mariadb

```
mysql -u user -p -h <IP-SERVEUR-MARIADB>
```



# Création du certificat CA (TLS/SSL)

## Note:

Le CN d'un certificat doit être **unique**. Si le CN est identique entre le client et le serveur il est impossible d'identifier une entité (le client) d'une autre (le serveur). Pour cela, on devra à chaque création de certificat, renseigner un CN (Common Name) différent, soit :

CA common Name : MariaDB\_admin  
Serveur common Name : [IP DU SERVEUR]  
Client common Name : MariaDB\_client

```
Country Name (2 letter code) [AU]:FR
State or Province Name (full name) [Some-State]:France
Locality Name (eg, city) []:Rochefort
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:Mariadb admin
Email Address []: _
```

# Création du certificat CA (TLS/SSL)

**a) Depuis le serveur, créez le répertoire /etc/mysql/ssl**

```
cd /etc/mysql  
sudo mkdir ssl  
cd ssl
```

**b) Créez la clé CA**

```
openssl genrsa 2048 > ca-key.pem
```

**c) Utilisez la clé CA pour générer le certificat CA pour Mariadb**

```
sudo openssl req -new -x509 -nodes -days 365000 -key  
ca-key.pem -out ca-cert.pem
```

/etc/mysql/ssl/ca-key.pem



Clé privé CA  
**ca-key.pem**

Fichier de clé pour  
l'autorité de certification  
(CA).

/etc/mysql/ssl/ca-cert.pem



Certificat CA  
**ca-cert.pem**

Fichier de certificat pour  
l'autorité de certification  
(CA).

# Création du certificat CA (TLS/SSL)

```
notBefore=May  1 16:02:30 2023 GMT  
notAfter=Sep  1 16:02:30 3022 GMT
```

## d) Vérifiez la validité du certificat CA

```
openssl x509 -noout -dates -in /etc/mysql/ssl/certificat.pem
```

## e) Vérifiez les droits du répertoire /etc/mysql/ssl

```
cd /etc/mysql  
ls -la
```

```
drwxr-xr-x  2 root root 4096 mai  1 18:02 ssl
```

## f) Vérifiez le propriétaire du répertoire /etc/mysql/ssl

```
cd ..  
ls -la
```

```
drwxr-xr-x  2 root root 4096 mai  1 18:02 ssl
```

## g) Vérifiez les droits d'accès et propriétaire du fichier /etc/mysql/ssl/ca-cert.pem

```
cd /etc/mysql/ssl  
ls -la
```

```
-rw-r--r--  1 mysql root 1119 mai  1 18:02 ca-cert.pem
```

```
sudo chmod 644 ./ca-cert.pem  
sudo chown mysql:root ./ca-cert.pem
```

# Création du certificat SSL Serveur

a) Toujours sur le serveur dans `/etc/mysql/ssl`, créez la certificat SSL serveur

```
openssl req -newkey rsa:2048 -days 365000 -nodes -keyout  
server-key.pem -out server-req.pem
```

b) Récupérez l'IP du serveur

```
hostname -I
```

⚠ **Votre CN (Common Name) doit IMPERATIVEMENT correspondre à votre nom de server. Nous prendrons l'IP, comme CN.**

```
Country Name (2 letter code) [AU]:FR  
State or Province Name (full name) [Some-State]:France  
Locality Name (eg, city) []:Rocheft  
Organization Name (eg, company) [Internet Widgits Pty Ltd]:  
Organizational Unit Name (eg, section) []:  
Common Name (e.g. server FQDN or YOUR name) []: 192.168.1.82  
Email Address []:
```

# Création du certificat SSL Serveur

**b) Supprimez la phrase secrète** associée à la clé privée server-key.pem

```
openssl rsa -in server-key.pem -out server-key.pem
```

**c) Signez le certificat serveur avec le certificat d'autorité et la clé CA**

```
openssl x509 -req -in server-req.pem -days 365000 -CA  
ca-cert.pem -CAkey ca-key.pem -set_serial 01 -out  
server-cert.pem
```

/etc/mysql/ssl/server-key.pem



Fichier de clé du serveur  
Mariadb

Clé privé serveur  
**server-key.pem**

/etc/mysql/ssl/server-cert.pem



Fichier de certificat du  
serveur Mariadb

Certificat serveur  
**server-cert.pem**

# Création du certificat SSL Serveur

## d) Confirmez l'emplacement du certificat serveur

```
ls -la /etc/mysql/ssl
```

```
-rw-r--r-- 1 mysql root 1119 mai  1 18:02 ca-cert.pem  
-rw-r--r-- 1 mysql root  977 mai  1 18:02 server-cert.pem  
-rw----- 1 mysql root 1704 mai  1 18:02 server-key.pem
```

## e) Vérifiez le CN de votre certificat serveur (server-cert.pem)

```
openssl x509 -noout -text -in /etc/mysql/ssl/server-cert.pem
```

```
Certificate:  
  Data:  
    Version: 1 (0x0)  
    Serial Number: 1 (0x1)  
    Signature Algorithm: sha256WithRSAEncryption  
    Issuer: CN = Mariadb_CA  
    Validity  
      Not Before: May  2 16:32:20 2023 GMT  
      Not After : Sep  2 16:32:20 3022 GMT  
    Subject: CN = 192.168.1.82
```

Le CN doit **IMPÉRATIVEMENT** être votre IP. Dans le cas contraire il faut réémettre un certificat pour le serveur.

# Création du certificat SSL Serveur

**f) Vérifiez le propriétaire (mysql:root) du certificat serveur et de la clé serveur**

```
ls -la
```

-rw-r--r--	1	mysql	root	977	mai	1	18:02	server-cert.pem
-rw-----	1	mysql	root	1704	mai	1	18:02	server-key.pem

Pour corriger

```
sudo chown mysql:root *.pem
```

**e) Vérifiez les droits d'accès du certificat serveur (-RW-R--R--) et de la clé serveur (-rw-----).**

Pour corriger:

```
sudo chmod 644 ./server-cert.pem
```

```
sudo chmod 600 ./server-key.pem
```

**f) Vérifiez la signature**

```
openssl verify -CAfile /etc/mysql/ssl/ca-cert.pem  
/etc/mysql/ssl/server-cert.pem
```

```
/etc/mysql/ssl/server-cert.pem: OK
```

# Configuration du serveur mariadb

## a) Modifiez le fichier de configuration du serveur Mariadb:

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



**[mysqld]**

```
...
log_warnings=9
log_error = /var/log/mysql/error.log
...
ssl-ca = /etc/mysql/ssl/ca-cert.pem
ssl-cert = /etc/mysql/ssl/server-cert.pem
ssl-key = /etc/mysql/ssl/server-key.pem
require-secure-transport = on
...
```

## b) Redémarrez le serveur Mariadb:

```
sudo systemctl restart mariadb.service
```

Si le serveur ne redémarre pas, lisez le message d'erreur de la commande

```
sudo systemctl status mariadb.service
```

Affichez les journaux d'erreur

```
cat /var/log/mysql/error.log
```



# Configuration du serveur mariadb

## c) créez le compte SQL

```
Mariadb -u root -p
```

```
CREATE DATABASE db_test ;  
CREATE USER 'admin'@'%' IDENTIFIED BY 'password' REQUIRE SSL;  
GRANT ALL PRIVILEGES ON db_test.* TO admin'@'%';  
FLUSH PRIVILEGES ;  
FLUSH SSL ;
```

'admin'@'%' signifie que l'utilisateur "admin" est autorisé à se connecter à la base de données à partir de n'importe quelle adresse IP ou nom d'hôte

La clause **REQUIRE SSL** spécifie que les connexions à ce compte doivent être établies en utilisant le protocole SSL. Si un client tente de se connecter à ce compte sans utiliser SSL ou TLS, la connexion sera refusée.

# Création du certificat SSL client

## a) Créez la certificat SSL client

```
sudo openssl req -newkey rsa:2048 -days 365000 -nodes  
-keyout client-key.pem -out client-req.pem
```

⚠ Renseignez un CN (Common Name) différent , soit Mariadb client:

```
Country Name (2 letter code) [AU]:FR  
State or Province Name (full name) [Some-State]:France  
Locality Name (eg, city) []:Rochefort  
Organization Name (eg, company) [Internet Widgits Pty Ltd]:  
Organizational Unit Name (eg, section) []:  
Common Name (e.g. server FQDN or YOUR name) []:Mariadb client  
Email Address []:
```

Le CN (Common Name) du client n'a pas obligatoirement besoin de porter le nom d'hôte du client.

# Création du certificat SSL client

**b) Supprimez la phrase secrète** associée à la clé privée client-key.pem

```
sudo openssl rsa -in client-key.pem -out client-key.pem
```

**c) Signez le certificat client avec le certificat d'autorité et la clé CA**

```
sudo openssl x509 -req -in client-req.pem -days 365000  
-CA ca-cert.pem -CAkey ca-key.pem -set_serial 01 -out  
client-cert.pem
```



client-cert.pem

Certificat client



client-key.pem

Clé privé client

# Copie des fichiers sur le client

## d) Copiez les fichiers suivants du serveur vers le client

**client-cert.pem**: certificat SSL du client

**client-key.pem**: clé privée du client

**ca-cert.pem**: autorité de certification (CA) qui a signé le certificat du client

## Depuis le client, utilisez la commande scp :

```
scp <UTILISATEUR>@<IP SERVEUR>:/chemin/fichier /chemin/client
```

où :

<UTILISATEUR> est le nom de l'utilisateur précédemment créé sur le serveur

<IP SERVEUR> est l'adresse IP du serveur

/chemin/fichier est le chemin **absolu** du fichier sur le serveur

/chemin/client est le chemin local où vous souhaitez copier le fichier

## Exemple :

```
scp nom_utilisateur@192.168.1.100:/etc/mysql/ssl/client-cert.pem /home/utilisateur/.mysql/
```

# Vérification coté client

```
-rw-rw-r-- 1 jf jf 977 mai 1 18:02 client-cert.pem  
-rw----- 1 jf jf 1704 mai 1 18:02 client-key.pem
```

## e) Toujours depuis le client, vérifier le propriétaire du certificat client

Si votre utilisateur courant (pas root) n'est pas le propriétaire, modifiez cela avec la commande :

```
sudo chown <USER>:<USER> *.pem
```

## f) Vérifiez les droits d'accès du certificat client (-RW-R--R--) et de la clé client (-rw-----).

Pour corriger:

```
sudo chmod 644 ./server-cert.pem
```

```
sudo chmod 600 ./server-key.pem
```

## g) Vérifiez la signature

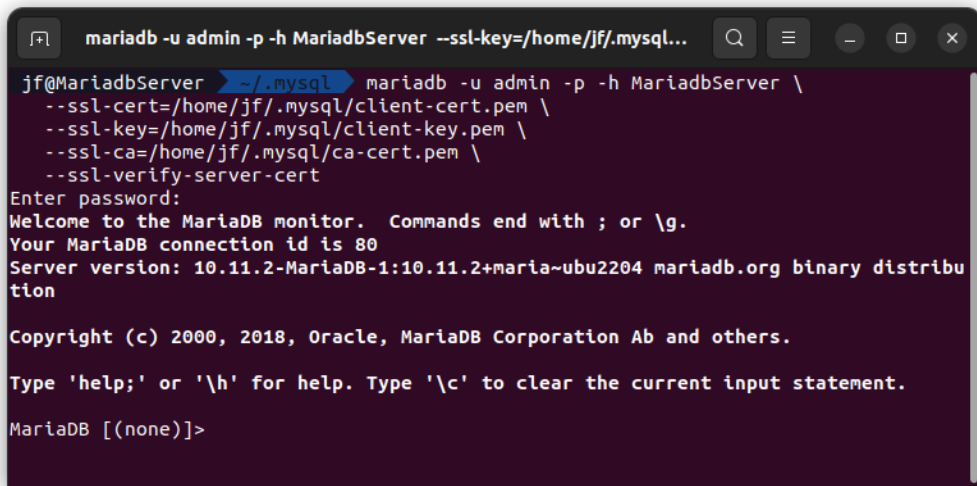
```
openssl verify -CAfile /etc/mysql/ssl/ca-cert.pem  
/etc/mysql/ssl/client-cert.pem
```

```
/etc/mysql/ssl/server-cert.pem: OK
```

# Configuration coté client

## a) Testez votre connexion

```
mariadb -u admin -p -h <VOTRE IP> \  
--ssl-cert=/home/jf/.mysql/client-cert.pem \  
--ssl-key=/home/jf/.mysql/client-key.pem \  
--ssl-ca=/home/jf/.mysql/ca-cert.pem \  
--ssl-verify-server-cert
```



```
mariadb -u admin -p -h MariadbServer --ssl-key=/home/jf/.mysql...  
jf@MariadbServer ~/.mysql$ mariadb -u admin -p -h MariadbServer \  
--ssl-cert=/home/jf/.mysql/client-cert.pem \  
--ssl-key=/home/jf/.mysql/client-key.pem \  
--ssl-ca=/home/jf/.mysql/ca-cert.pem \  
--ssl-verify-server-cert  
Enter password:  
Welcome to the MariaDB monitor.  Commands end with ; or \g.  
Your MariaDB connection id is 80  
Server version: 10.11.2-MariaDB-1:10.11.2+Maria-ubu2204 mariadb.org binary distribu  
tion  
  
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)]>
```

# Configuration coté client

## a) Editez le fichier `~/ .mysql/my .cnf`

```
nano ~/ .mysql/my .cnf
```

## b) Ajoutez ces lignes au groupe et sauvegardez

```
#ssl  
ssl-ca=/home/<VOTRE UTILISATEUR>/ .mysql/ca-cert .pem  
ssl-cert=/home/<VOTRE UTILISATEUR>/ .mysql/client-cert .pem  
ssl-key=/home/<VOTRE UTILISATEUR>/ .mysql/client-key .pem
```

## c) Testez votre configuration client

```
mariadb -u admin -p -h MariadbServer
```

Les paramètres `ssl-ca`, `ssl-cert` et `ssl-key` sont maintenant renseignés par le fichier `my.cnf`

# Configuration de Dbeaver



DBeaver

Configuration de la connexion "MariadbServer"

Paramètres de connexion  
MariaDB paramètres de connexion

Paramètres de connexion

Général Propriétés du pilote SSH Proxy SSL

Server Host: MariadbServer Port: 3306

Database:

Authentification (Database Native)

Nom d'utilisateur: admin

Mot de passe: ..... ☒ Enregistrer

Advanced

Server Time Zone: Auto-detect

Local Client:

[Vous pouvez utiliser des variables dans les paramètres de connexion.](#)

Driver name: MariaDB Driver Settings License du pilote

Test de la connexion ... Annuler OK

Général Propriétés du pilote SSH Proxy SSL

Server

Server Host: 192.168.1.82 Port: 3306

Database:

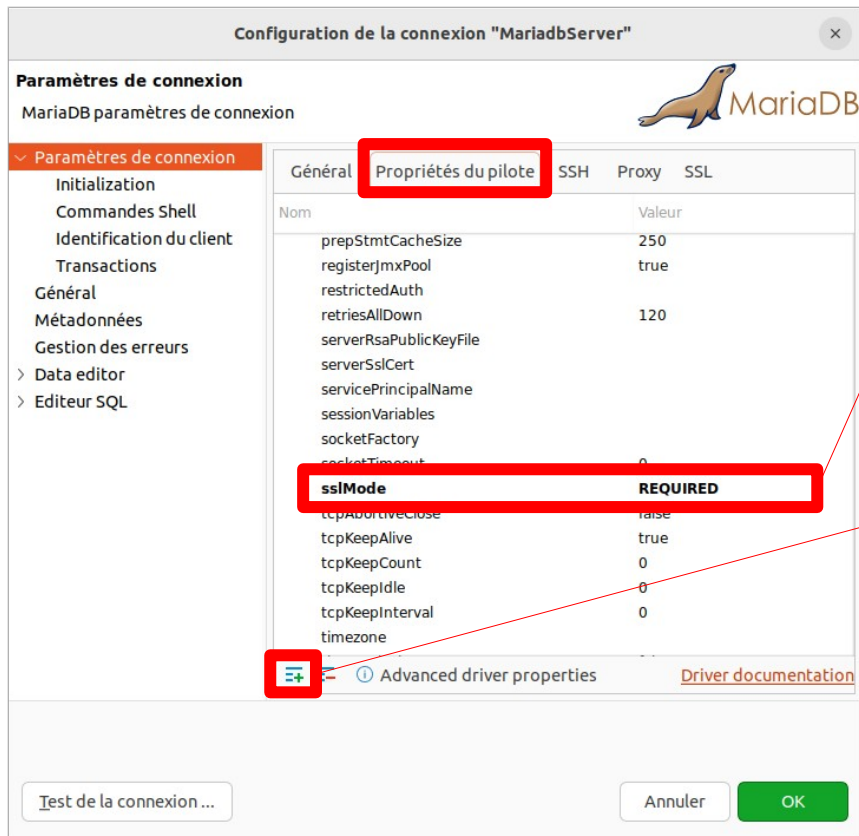
Authentification (Database Native)

Nom d'utilisateur: admin

Mot de passe: ..... ☒ Enregistrer



# Configuration de Dbeaver

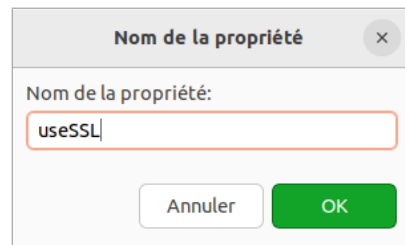


## Dans l'onglet « Propriétés du pilote »

1) Vérifiez que le paramètre sslMode soit sur REQUIRED

**sslMode** **REQUIRED**

2) Ajoutez le paramètre useSSL



3) Donnez lui la valeur TRUE

**useSSL** **TRUE**

# Configuration de Dbeaver



DBeaver

Configuration de la connexion "MariadbServer"

Paramètres de connexion  
MariaDB paramètres de connexion

Paramètres de connexion

- Initialisation
- Commandes Shell
- Identification du client
- Transactions
- Général
- Métadonnées
- Gestion des erreurs
- > Data editor
- > Editeur SQL

Général Propriétés du pilote SSH Proxy **SSL**

☒ Utiliser SSL Profile:

All SSL parameters are optional.  
You must specify SSL certificates if they are required by your server configuration.

Parameters

CA Certificate:

Client Certificate:

Client Private Key:

Cipher suites (optional):

Advanced

Require SSL: ☒

Verify server certificate: ☒

Allow public key retrieval: ☐

Test de la connexion ...

Annuler OK

## Dans l'onglet «SSL»

1) Dans le champs CA certificate  
Ajoutez le fichier **ca-cert.pem**

2) Dans le champs « Client certificate »  
Ajoutez le fichier **client-cert.pem**

3) Dans le champs « Client Private Key»  
Ajoutez le fichier **client-key.pem**

4) Dans l'encart « Advanced »  
Cochez les cases **Require SSL** et **Vérify server certificate**

# Configuration de Dbeaver



DBeaver

Testez la connexion

Configuration de la connexion "MariadbServer"

Paramètres de connexion  
MariaDB paramètres de connexion

Paramètres de connexion

- Initialisation
- Commandes Shell
- Identification du client
- Transactions
- Général
- Métadonnées
- Gestion des erreurs
- > Data editor
- > Editeur SQL

Général Propriétés du pilote SSH Proxy **SSL**

☒ Utiliser SSL Profile:

All SSL parameters are optional.  
You must specify SSL certificates if they are required by your server configuration.

Parameters

CA Certificate:

Client Certificate:

Client Private Key:

Cipher suites (optional):

Advanced

Require SSL: ☒

Verify server certificate: ☒

Allow public key retrieval: ☐

Test de la connexion ... Annuler **OK**

Enregistrer

# Et voila !!!

