Configure ssh jump server to use SSL for Mariadb

736 Manu Chacko May 1, 2025 Tweaks & Configuration 3074

How to configure Ezeelogin PAM solutions to use SSL for Mariadb

Overview: This article explains how to configure SSL for MariaDB on the Ezeelogin SSH jump server. It includes checking the SSL status, generating certificates, configuring Ezeelogin, and verifying the connection to ensure secure database access.

Mysql-SSL setup on Mariadb Server

Step 1: Check the Current SSL/TLS Status

Log into a Mariadb server with following command

```
root@gateway:~# mysql -u root -p
```

Show the state of the SSL/TLS variables by typing:

```
MariaDB [(none)]> show variables like '%ssl%';
--------------
show variables like '%ssl%'
--------------

+--------------------+----------------------------+
| Variable_name      | Value                      |
+--------------------+----------------------------+
| have_openssl       | YES                        |
| have_ssl           | DISABLED                   |
| ssl_ca             |                            |
| ssl_capath         |                            |
| ssl_cert           |                            |
| ssl_cipher         |                            |
| ssl_crl            |                            |
| ssl_crlpath        |                            |
| ssl_key            |                            |
| version_ssl_library | OpenSSL 3.0.2 15 Mar 2022 |
+--------------------+----------------------------+
10 rows in set (0.004 sec)
```

The **have_ssl** variable is marked as DISABLED. This means that SSL functionality has been compiled into the server, but it is not yet enabled

Step 2: Generate SSL/TLS Certificates and Keys

Create a clean environment

```
root@gateway:~# mkdir /etc/certs && cd /etc/certs
```

Create the CA certificate

```
root@gateway:~#openssl genrsa 2048 > ca-key.pem

root@gateway:~#openssl req -new -x509 -nodes -days 3600 \
-key ca-key.pem -out ca.pem
```

Create the server certificate, remove passphrase, and sign it

```
root@gateway:~#openssl req -newkey rsa:2048 -days 3600 \
-nodes -keyout server-key.pem -out server-req.pem


root@gateway:~#openssl rsa -in server-key.pem -out server-key.pem

root@gateway:~#openssl x509 -req -in server-req.pem -days 3600 \
-CA ca.pem -CAkey ca-key.pem -set_serial 01 -out server-cert.pem
```

Create the client certificate, remove passphrase, and sign it

```
root@gateway:~#openssl req -newkey rsa:2048 -days 3600 \
-nodes -keyout client-key.pem -out client-req.pem

root@gateway:~#openssl rsa -in client-key.pem -out client-key.pem

root@gateway:~#openssl x509 -req -in client-req.pem -days 3600 \
-CA ca.pem -CAkey ca-key.pem -set_serial 01 -out client-cert.pem
```

After generating the certificates, verify them:

```
root@gateway:~#  openssl verify -CAfile ca.pem server-cert.pem client-cert.pem


output

server-cert.pem: OK
client-cert.pem: Ok
```

Enable SSL for Mariadb

Modify the Mariadb configuration file **'/etc/mysql/mariadb.conf.d/50-server.cnf'**

In the '**[mysqld]**' section, paste the configuration below.

```
root@gateway:~# vi /etc/mysql/mariadb.conf.d/50-server.cnf

ssl-ca=/etc/certs/ca.pem

ssl-cert=/etc/certs/server-cert.pem

ssl-key=/etc/certs/server-key.pem
```

Restart the MySQL service

```
root@gateway:~# systemctl restart mariadb
```

After restarting, open up a new MySQL session using the same command as before.

```
root@gateway:~# mysql -u root -p
```

Check state of the SSL/TLS variables by typing:

```
MariaDB [(none)]> show variables like '%ssl%';
--------------
show variables like '%ssl%'
--------------

+--------------------+--------------------------+
| Variable_name      | Value                    |
+--------------------+--------------------------+
| have_openssl       | YES                      |
| have_ssl           | YES                      |
| ssl_ca             | /etc/certs/ca.pem        |
| ssl_capath         |                          |
| ssl_cert           | /etc/certs/server-cert.pem |
| ssl_cipher         |                          |
| ssl_crl            |                          |
| ssl_crlpath        |                          |
| ssl_key            | /etc/certs/server-key.pem  |
| version_ssl_library | OpenSSL 3.0.2 15 Mar 2022 |
+--------------------+--------------------------+
10 rows in set (0.004 sec)
```

The **have_openssl** and **have_ssl** variables read **"YES"** instead of "**DISABLED**" this time.

Now you can login to Mariadb server with following command and grant Ezeelogin user to access the Ezeelogin database. you can refer the article to retrieve Ezeelogin database credentials.

Replace **ezlogin_databasename,ezlogin_db_username** and **ez_db_password** with your Ezeelogin database username.

```
root@gateway:~# mysql -u root -p

[Enter password]

MariaDB [(none)]> create user 'ezlogin_db_username'@'127.0.0.1' identified by 'ez_db_password';

MariaDB [(none)]> grant all on ezlogin_databasename.* to 'ezlogin_db_username'@'127.0.0.1' with grant option;

MariaDB [(none)]> flush privileges;

MariaDB [(none)]> exit
```

Check the connection details by the following command :

```
root@gateway ~]# mysql -u ezlogin_db_username -p -h 127.0.0.1  --ssl-ca=/etc/certs/ca.pem --ssl-cert=/etc/certs/client-cert.pem --ssl-key=/etc/certs/client-key.pem

example :
```

```
root@gateway~]# mysql -u ezlogin_xxxx -p -h 127.0.0.1 --ssl-ca=/etc/certs/ca.pem --ssl-cert=/etc/certs/client-cert.pem --ssl-key=/etc/certs/client-key.pem
```

In Case the certificate verification has been failed, refer **SSL certificate failed with MYSQL SSL**

```
mysql> \s

---------------

. . .

SSL: Cipher in use is DHE-RSA-AES256-SHA

. . .

Connection: 127.0.0.1 via TCP/IP

. . .

----------------
```

SSL cipher is displayed, indicating that SSL is being used to secure our connection.

Step 3: Configure ezeelogin jump server to use SSL for Mariadb

Add **mysql_ssl_key, mysql_ssl_cert, mysql_ssl_ca** and change **db_hos, db_port** to **/usr/local/etc/ezlogin/ez.conf** as follows

Edit the **/usr/local/etc/ezlogin/ez.conf** file add the following

```
root@gateway:~# vi /usr/local/etc/ezlogin/ez.conf

#Add the following

system_folder /var/www/ezlogin/
force_https no
uri_path /ezlogin/
db_host 127.0.0.1
db_port 3306
db_name ezlogin_qzms
db_user ezlogin_edcjwz
db_pass dsH)$s5xAE[QgFms
db_prefix aqvo_
cookie_encryption_key ASvs8^pnu^^X9
cookie_name lcrrfs
cookie_path /ezlogin/
www_folder /var/www/html/ezlogin/
admin_user admin
mysql_encrypt yes
mysql_ssl_key /etc/certs/client-key.pem
mysql_ssl_cert /etc/certs/client-cert.pem
mysql_ssl_ca /etc/certs/ca.pem
mysql_ssl_capath /etc/certs/
mysql_ssl_verify no
```

Make sure that you have changed db_port to 3306 & db_host to 127.0.0.1 of your host

Step 4: Change the bind address & allow the Ezeelogin jump server user to access the database.

Edit the /etc/mysql/mariadb.conf.d/50-server.cnf & change bind-address

```
root@gateway:~# vi /etc/mysql/mariadb.conf.d/50-server.cnf

Change bind-address to host ip or 0.0.0.0

bind-address 0.0.0.0
```

Restart the Mariadb service

```
root@gateway:~# systemctl restart mariadb
```

Step 5: Check and correct the permission of /etc/certs directory and client-key.pem

```
root@gateway:~# chmod 755 /etc/certs/


root@gateway:~# ls -ld /etc/certs/
drwxr-xr-x 2 root root 4096 Sep 20 15:51 /etc/certs/

:~# chmod 644 /etc/certs/client-key.pem


root@gateway:~# ls -ld /etc/certs/client-key.pem
-rw-r--r-- 1 mysql mysql 1705 Mar 29 2023 /etc/certs/client-key.pem
```

You have successfully configured SSL for MariaDB on the Ezeelogin SSH jump server, enhancing the security of your database connections and ensuring that sensitive data is encrypted during transmission.

Make sure to grant the secondary IP to access primary node and the primary IP to the secondary node when configuring the cluster. Replace the secondary IP and primary IP with the appropriate IPs.

```
root@gateway:~# mysql -u root -p

[Enter password]

MariaDB [(none)]> create user 'ezlogin_db_username'@'secondary IP or primary IP' identified by 'ez_db_password';

MariaDB [(none)]> grant all on ezlogin_databasename.* to 'ezlogin_db_username'@'secondary IP or primary IP' with grant option;

MariaDB [(none)]> flush privileges;

MariaDB [(none)]> exit
```

If you have any difficulties please **contact support**

Related Articles:

Configure ssh jump server to use SSL for MySQL

Install Master/Slave Ezeelogin with MySQL SSL

Unable to access GUI while using MySQL SSL

failed to connect to database: Error: TLS/SSL error: Permission denied

Online URL: https://www.ezeelogin.com/kb/article/configure-ssh-jump-server-to-use-ssl-for-mariadb-736.html