How to connect mysql client over SSH tunnel to Mariadb server

For setting up your working Mariadb server, read this howto.

You may want to connect clients to the database server over network securely. Instead of using TLS (which is available natively) for this, the same can also be achieved by tunneling over SSH by following the below steps.

On the database server

1. Ensure there is no blocking by firewall for the ports/protocols/hosts.. The default Mariadb port is TCP 3306.
2. Ensure that the SKIP="..skip-networking…" line in /etc/rc.d/rc.mysqld is commented out. This will allow external client connections to the server.
3. Identify a user who will be the SSH tunnel user. This refers to the Linux user. You may use the command 'adduser <username>' to create a new user.
4. Edit /etc/ssh/sshd_config, if necessary, to ensure that key based authentication is allowed for this user. The /etc/rc.sshd service will of course have to be restarted if any edits had been made.

On the client machine

Generate keys for authentication using ssh-keygen without a passphrase as follows. You may use RSA also. The below command will create a private and public key after giving you file name option. Just hit enter to avoid using a passphrase. Create and login as a similar tunnel user when doing this.

```
ssh-keygen -t ecdsa -b 521 -C my-comments
```

The public key with filename extension .pub should be installed in the database server user's authorized_keys. The contents of the .pub file can be copy pasted or you can use a command line utility like ssh-copy-id as shown below. Never copy the private key to other server.

```
ssh-copy-id -i ~/.ssh/mykey user@database-server-host
```

It is recommended to use autossh, which is available in slackbuilds, to make persistent connections. Once autossh is installed, you can now create the tunnel with a command like below. The port 334 on the client machine can be any unassigned port you choose.

```
autossh -M 10984 -o "PubkeyAuthentication=yes" -o "PasswordAuthentication=no" -o "ServerAliveInterval 60" -o "ServerAliveCountMax 3" -i ~/.ssh/mykey -L 334:127.0.0.1:3306 user@database-server-host
```

Connect the client

```
mysql -u root -p  --host=127.0.0.1 --port=334
```

You thus have a secure ssh tunneling for mysql client now.

Sources

Adapted from original source: https://mariadb.com/resources/blog/connecting-to-mariadb-through-an-ssh-tunnel/

Originally written by mahafyi

howtos, mariadb, mysql, tunneling, ssh, tunnel

From:
https://docs.slackware.com/ - **SlackDocs**

Permanent link:
https://docs.slackware.com/howtos:misc:connect_client_over_ssh_tunnel_to_mariadb

Last update: **2019/06/26 01:51 (UTC)**