Troubleshooting Connection Issues

If you are completely new to MariaDB and relational databases, you may want to start with the MariaDB Primer. Also, make sure you understand the connection parameters discussed in the Connecting to MariaDB article.

There are a number of common problems that can occur when connecting to MariaDB.

Server not running in specified location

If the error you get is something like:

```
mysql -uname -p
ERROR 2002 (HY000): Can't connect to local MySQL server through
  socket '/var/run/mysqld/mysqld.sock' (2 "No such file or directory")
```

or

```
mysql -uname -p --port=3307 --protocol=tcp
ERROR 2003 (HY000): Can't connect to MySQL server on 'localhost' (111 "Connection refused")
```

the server is either not running, or not running on the specified port, socket or pipe. Make sure you are using the correct host, port, pipe, socket and protocol options, or alternatively, see Getting, Installing and Upgrading MariaDB, Starting and Stopping MariaDB or Troubleshooting Installation Issues.

The socket file can be in a non-standard path. In this case, the `socket` option is probably written in the my.cnf file. Check that its value is identical in the [mysqld] and [client] sections; if not, the client will look for a socket in a wrong place.

If unsure where the Unix socket file is running, it's possible to find this out, for example:

```
netstat -ln | grep mysqld
unix  2      [ ACC ]     STREAM     LISTENING     33209505 /var/run/mysqld/mysqld.sock
```

Unable to connect from a remote location

By default, the MariaDB server does not accept connections from a remote client and has to be configured to permit these. See Configuring MariaDB for Remote Client Access

Authentication problems

Authentication is granted to a particular username/host combination. `user1'@'localhost'` , for example, is not the same as `user1'@'166.78.144.191'` . See the GRANT article for details on granting permissions.

Passwords are hashed with PASSWORD function. If you have set a password with the SET PASSWORD statement, or used INSERT or UPDATE to update the permissions table directly, the PASSWORD function must be used at the same time. For example, `SET PASSWORD FOR 'bob'@'%.loc.gov' = PASSWORD('newpass')` rather than just `SET PASSWORD FOR 'bob'@'%.loc.gov' = 'newpass'` ;

If grant tables have been changed directly, the new passwords or authentication data will not immediately be active. A FLUSH PRIVILEGES statement, or the flush-privileges mysqladmin option must be run in order for the changes to take effect.

Problems exporting query results

If you can run regular queries, but get an authentication error when running the SELECT ... INTO OUTFILE, SELECT ... INTO DUMPFILE or LOAD DATA INFILE statements, you do not have permission to write files to the server. This requires the FILE privilege. See the GRANT article.

Access to the server, but not to a database

If you can connect to the server, but not to a database, for example:

```
MariaDB [(none)]> use test;
ERROR 1044 (42000): Access denied for user 'ian'@'localhost' to database 'test'
```

or can connect to a particular database, but not another, for example `mysql -u name db1` works but not `mysql -u name db2` , you have not been granted permission for the particular database. See the GRANT article.

Option files and environment variables

It's possible that option files or environment variables may be providing incorrect connection parameters. Check the values provided in any option files read by the client you are using (see mysqld Configuration Files and Groups and the documentation for the particular client you're using (see Clients and Utilities.

Option files can usually be suppressed with `no-defaults` option, for example:

```
mysqlimport --no-defaults ...
```