Getting Started With MariaDB MaxScale
Read/Write Splitting with MySQL Replication

Mark Riddoch

Last Updated: 19th November 2014

Contents

Environment & Solution Space

This document is designed as a quick introduction to setting up MaxScale in an environment in which you have a MySQL Replication Cluster with one master and multiple slave servers. The object of this tutorial is to have a system that appears to the clients of MaxScale as if there is a single database behind MaxScale. MaxScale will split the statements such that write statements will be sent to the current master server in the replication cluster and read statements will be balanced across a number of the slave statements.

The process of setting and configuring MaxScale will be covered within this document. However the installation and configuration of the MySQL Replication subsystem will not be covered nor will any discussion of installation management tools to handle automated or semi-automated failover of the replication cluster.

This tutorial will assume the user is running from one of the binary distributions available and has installed this in the default location. Building from source code in GitHub is covered in guides elsewhere as is installing to non-default locations.

Process

The steps involved in creating a system from the binary distribution of MaxScale are:
- Install the package relevant to your distribution
- Create the required users in your MariaDB or MySQL Replication cluster
- Create a MaxScale configuration file

Installation

The precise installation process will vary from one distribution to another details of what to do with the RPM and DEB packages can be found on the download site when you select the distribution you are downloading from. The process involves setting up your package manager to include the MariaDB repositories and then running the package manager for your distribution, RPM or apt-get.

Upon successful completion of the installation command you will have MaxScale installed and ready to be run but without a configuration. You must create a configuration file before you first run MaxScale.

Creating Database Users

MaxScale needs to connect to the backend databases and run queries for two reasons; one to determine the current state of the database and the other to retrieve the user information for the database cluster. This may be done either using two separate usernames or with a single user.

The first user required must be able to select data from the table mysql.user, to create this user follow the steps below.

1. Connect to the current master server in your replication tree as the root user

2. Create the user, substituting the username, password and host on which maxscale runs within your environment

```
MariaDB [(none)]> create user 'username'@'maxscalehost' identified by
'password';
Query OK, 0 rows affected (0.00 sec)
```

3. Grant select privileges on the mysql.user table.

```
MariaDB [(none)]> grant SELECT on mysql.user to 'username'@'maxscalehost';
Query OK, 0 rows affected (0.03 sec)
```

Additionally, GRANT SELECT on the mysql.db table and SHOW DATABASES privileges are required in order to load databases name and grants suitable for database name authorization.

```
MariaDB [(none)]> GRANT SELECT ON mysql.db TO 'username'@'maxscalehost';
Query OK, 0 rows affected (0.00 sec)
MariaDB [(none)]> GRANT SHOW DATABASES ON *.* TO 'username'@'maxscalehost';
Query OK, 0 rows affected (0.00 sec)
```

The second user is used to monitored the state of the cluster. This user, which may be the same username as the first, requires permissions to access the various sources of monitoring data. In order to monitor a replication cluster this user must be granted the roles REPLICATION SLAVE and REPLICATION CLIENT

```
MariaDB [(none)]> grant REPLICATION SLAVE on *.* to 'username'@'maxscalehost';
Query OK, 0 rows affected (0.00 sec)
```

```
MariaDB [(none)]> grant REPLICATION CLIENT on *.* to 'username'@'maxscalehost';
Query OK, 0 rows affected (0.00 sec)
```

If you wish to use two different usernames for the two different roles of monitoring and collecting user information then create a different username using the first two steps from above.

Creating Your MaxScale Configuration

MaxScale configuration is held in an ini file that is located in the file MaxScale.cnf in the directory $MAXSCALE_HOME/etc, if you have installed in the default location then this file is available in /usr/local/skysql/maxscale/etc/MaxScale.cnf. This is not created as part of the installation process and must be manually created. A template file does exist within this directory that may be use as a basis for your configuration.

A global, maxscale, section is included within every MaxScale configuration file; this is used to set the values of various MaxScale wide parameters, perhaps the most important of these is the number of threads that MaxScale will use to execute the code that forwards requests and handles responses for clients.

```
[maxscale]
threads=4
```

The first step is to create a service for our Read/Write Splitter. Create a section in your MaxScale.ini file and set the type to service, the section names are the names of the services themselves and should be meaningful to the administrator. Names may contain whitespace.

```
[Splitter Service]
type=service
```

The router for we need to use for this configuration is the readwritesplit module, also the services should be provided with the list of servers that will be part of the cluster. The server names given here are actually the names of server sections in the configuration file and not the physical hostnames or addresses of the servers.

```
[Splitter Service]
type=service
router=readwritesplit
servers=dbserv1, dbserv2, dbserv3
```

The final step in the service sections is to add the username and password that will be used to populate the user data from the database cluster. There are two options for representing the password, either plain text or encrypted passwords may be used. In order to use encrypted passwords a set of keys must be generated that will be used by the encryption and decryption process. To generate the keys use the `maxkeys` command and pass the name of the secrets file in which the keys are stored.

```
% maxkeys /usr/local/skysql/maxscale/etc/.secrets
%
```

Once the keys have been created the `maxpasswd` command can be used to generate the encrypted password.

```
% maxpasswd plainpassword
96F99AA1315BDC3604B006F427DD9484
%
```

The username and password, either encrypted or plain text, are stored in the service section using the user and passwd parameters.

```
[Splitter Service]
type=service
router=readwritesplit
servers=dbserv1, dbserv2, dbserv3
user=maxscale
passwd=96F99AA1315BDC3604B006F427DD9484
```

This completes the definitions required by the service, however listening ports must be associated with the service in order to allow network connections. This is done by creating a series of listener sections. This section again is named for the convenience of the administrator and should be of type listener with an entry labelled service which contains the name of the service to associate the listener with. A service may have multiple listeners.

```
[Splitter Listener]
type=listener
service=Splitter Service
```

A listener must also define the protocol module it will use for the incoming network protocol, currently this should be the MySQLClient protocol for all database listeners. The listener may then supply a network port to listen on and/or a socket within the file system.

```
[Splitter Listener]
type=listener
service=Splitter Service
protocol=MySQLClient
port=3306
socket=/tmp/ClusterMaster
```

An address parameter may be given if the listener is required to bind to a particular network address when using hosts with multiple network addresses. The default behaviour is to listen on all network interfaces.

The next stage is the configuration is to define the server information. This defines how to connect to each of the servers within the cluster, again a section is created for each server, with the type set to server, the network address and port to connect to and the protocol to use to connect to the server. Currently the protocol module for all database connections in MySQLBackend.

```
[dbserv1]
type=server
address=192.168.2.1
port=3306
protocol=MySQLBackend

[dbserv2]
type=server
address=192.168.2.2
port=3306
protocol=MySQLBackend

[dbserv3]
type=server
address=192.168.2.3
port=3306
protocol=MySQLBackend
```

In order for MaxScale to monitor the servers using the correct monitoring mechanisms a section should be provided that defines the monitor to use and the servers to monitor. Once again a section is created with a symbolic name for the monitor, with the type set to monitor. Parameters are added for the module to use, the list of servers to monitor and the username and password to use when connecting to the the servers with the monitor.

```
[Replication Monitor]
type=monitor
module=mysqlmon
servers=dbserv1, dbserv2, dbserv3
user=maxscale
passwd=96F99AA1315BDC3604B006F427DD9484
```

As with the password definition in the server either plain text or encrypted passwords may be used.

The final stage in the configuration is to add the option service which is used by the `maxadmin` command to connect to MaxScale for monitoring and administration purposes. This creates a service section and a listener section.

```
[CLI]
type=service
router=cli

[CLI Listener]
type=listener
service=CLI
protocol=maxscaled
address=localhost
port=6603
```

In the case of the example above it should be noted that an address parameter has been given to the listener, this limits connections to `maxadmin` commands that are executed on the same machine that hosts MaxScale.

Starting MaxScale

Upon completion of the configuration process MaxScale is ready to be started for the first time. This may either be done manually by running the maxscale command or via the service interface.

```
% maxscale
```

or

```
% service maxscale start
```

Check the error log in /usr/local/skysql/maxscale/log to see if any errors are detected in the configuration file and to confirm MaxScale has been started. Also the maxadmin command may be used to confirm that MaxScale is running and the services, listeners etc have been correctly configured.

```
% maxadmin -pskysql list services
Services.
-----------------------+--------------------+--------+---------------
Service Name           | Router Module      | #Users | Total Sessions
-----------------------+--------------------+--------+---------------
Splitter Service       | readwritesplit     |     1 |    1
CLI                    | cli                |     2 |    2
-----------------------+--------------------+--------+---------------
% maxadmin -pskysql list servers
Servers.
-----------------+---------------+-------+------------+-------------------
Server           | Address       | Port  | Connections | Status
-----------------+---------------+-------+------------+-------------------
dbserv1          | 192.168.2.1   | 3306  |           0 | Running, Slave
dbserv2          | 192.168.2.2   | 3306  |           0 | Running, Master
dbserv3          | 192.168.2.3   | 3306  |           0 | Running, Slave
-----------------+---------------+-------+------------+-------------------
% maxadmin -pskysql list listeners
Listeners.
-------------------+------------------+---------------+-------+-------
Service Name       | Protocol Module  | Address       | Port  | State
-------------------+------------------+---------------+-------+-------
Splitter Service   | MySQLClient      | *             | 3306  | Running
CLI                | maxscaled        | localhost     | 6603  | Running
-------------------+------------------+---------------+-------+-------
%
```

MaxScale is now ready to start accepting client connections and routing them to the master or slaves within your cluster. Other configuration options are available that can alter the criteria used for routing, these include monitoring the replication lag within the cluster and routing only to slaves that are within a predetermined delay from the current master or using weights to obtain unequal balancing operations. These options may be found in the MaxScale Configuration Guide. More detail on the use of maxadmin can be found in the document "MaxAdmin - The MaxScale Administration & Monitoring Client Application".