w: www.mariadb.com

e: info@mariadb.com

Getting Started With MariaDB MaxScale

Common Administration Tasks

The purpose of this tutorial is to introduce the MaxScale Administrator to a few of the common administration tasks that need to be performed with MaxScale. It is not intended as a reference to all the tasks that may be performed, more this is aimed as an introduction for administrators who are new to MaxScale.

Starting MaxScale

There are several ways to start MaxScale, the most convenient mechanism is probably using the Linux service interface. When a MaxScale package is installed the package manager will also installed a script in /etc/init.d which may be used to start and stop MaxScale either directly or via the service interface.

```
$ service maxscale start
```

or

```
$ /etc/init.d/maxscale start
```

It is also possible to start MaxScale by executing the maxscale command itself, in this case you must ensure that the environment is correctly setup or command line options are passed. The major elements to consider are the correct setting of the MAXSCALE_HOME directory and to ensure that LD_LIBRARY_PATH. The LD_LIBRARY_PATH should include the lib directory that was installed as part of the MaxScale installation, the MAXSCALE_HOME should point to /usr/local/skysql/maxscale if a default installation has been created or to the directory this was relocated to. Running the executable $MAXSCALE_HOME/bin/maxscale will result in MaxScale running as a daemon process, unattached to the terminal in which it was started and using configuration files that it finds in the $MAXSCALE_HOME directory.

Options may be passed to the MaxScale binary that alter this default behaviour, this options are documented in the table below.

Switch	Long Option	Description
-d	--nodaemon	Run MaxScale attached to the terminal rather than as a daemon process. This is useful for debugging

w: www.mariadb.com

e: info@mariadb.com

		purposes.
-c	--homedir=	Ignore the environment variable MAXSCALE_HOME and use the supplied argument instead.
-f	--config=	Use the filename passed as an argument instead of looking in $MAXSCALE_HOME/etc/MaxScale.cnf
-l<file>| <shm>	--log=	Control where logs are written for the debug and trace level log messages. the default is to write these to a shared memory device, however using the -lfile or --log=file option will forced these to be written to regular files.
-v	--version	Print version information for MaxScale
-?	--help	Print usage information for MaxScale

Stopping MaxScale

There are numerous ways in which MaxScale can be stopped; using the service interface, killing the process or by use of the maxadmin utility.

Stopping MaxScale with the service interface is simply a case of using the service stop command or calling the init.d script with the stop argument.

```
$ service maxscale stop
```

or

```
$ /etc/init.d/maxscale stop
```

MaxScale will also stop gracefully if it received a hangup signal, to find the process id of the MaxScale server use the ps command or read the contents of the maxscale.pid file located in the same directory as the logs.

```
$ kill -HUP `cat $MAXSCALE_HOME/log/maxscale.pid`
```

w: www.mariadb.com

e: info@mariadb.com

In order to shutdown MaxScale using the maxadmin command you may either connect with maxadmin in interactive mode or pass the “shutdown maxscale” command you wish to execute as an argument to maxadmin.

```
$ maxadmin -pskysql shutdown maxscale
```

Checking The Status Of The MaxScale Services

It is possible to use the maxadmin command to obtain statistics regarding the services that are configured within your MaxScale configuration file. The maxadmin command “list services” will give very basic information regarding the services that are define. This command may be either run in interactive mode or passed on the maxadmin command line.

```
$ maxadmin -pskysql
MaxScale> list services
Services.
--------------------------+----------------------+--------+---------------
Service Name              | Router Module        | #Users | Total Sessions
--------------------------+----------------------+--------+---------------
RWSplitter                | readwritesplit       |      2 |     4
Cassandra                 | readconncouter       |      1 |     1
CLI                       | cli                  |      2 |     2
--------------------------+----------------------+--------+---------------

MaxScale>
```

It should be noted that network listeners count as a user of the service, therefore there will always be one user per network port in which the service listens. More detail can be obtained by use of the “show service” command which is passed a service name.

What Clients Are Connected To MaxScale

To determine what client are currently connected to MaxScale you can use the “list clients” command within maxadmin. This will give you IP address and the ID’s of the DCB and session for that connection. As with any maxadmin command this can be passed on the command line or typed interactively in maxadmin.

```
$ maxadmin -pskysql list clients
```

w: www.mariadb.com

e: info@mariadb.com

```
Client Connections
-----------------+------------------+----------------------+------------
 Client          | DCB              | Service              | Session
-----------------+------------------+----------------------+------------
 127.0.0.1       |   0x7fe694013410 | CLI                  | 0x7fe69401ac10
-----------------+------------------+----------------------+------------

$
```

Rotating Log Files

MaxScale write log data into four log files with varying degrees of detail. With the exception of the error log, which can not be disabled, these log files may be enabled and disabled via the maxadmin interface or in the configuration file. The default behaviour of MaxScale is to grow the log files indefinitely, the administrator must take action to prevent this.

It is possible to rotate either a single log file or all the log files with a single command. When the logfile is rotated, the current log file is closed and a new log file, with an increased sequence number in its name, is created. Log file rotation is achieved by use of the “flush log” or “flush logs” command in maxadmin.

```
$ maxadmin -pskysql flush logs
```

Flushes all of the logs, whereas an individual log may be flushed with the “flush log” command.

```
$ maxadmin -pskysql
MaxScale> flush log error
MaxScale> flush log trace
MaxScale>
```

This may be integrated into the Linux logrotate mechanism by adding a configuration file to the /etc/logrotate.d directory. If we assume we want to rotate the log files once per month and wish to keep 5 log files worth of history, the configuration file would look like the following.

```
/usr/local/skysql/maxscale/log/*.log {
monthly
rotate 5
missingok
nocompress
sharedscripts
postrotate
```

w: www.mariadb.com

e: info@mariadb.com

```
# run if maxscale is running
if test -n "`ps acx|grep maxscale`"; then
/usr/local/skysql/maxscale/bin/maxadmin -pskysql flush logs
fi
endscript
}
```

One disadvantage with this is that the password used for the maxadmin command has to be embedded in the log rotate configuration file. MaxScale will also rotate all of its log files if it receives the USR1 signal. Using this the logrotate configuration script can be rewritten as

```
/usr/local/skysql/maxscale/log/*.log {
monthly
rotate 5
missingok
nocompress
sharedscripts
postrotate
kill -USR1 `cat /usr/local/skysql/maxscale/log/maxscale.pid`
endscript
}
```

Taking A Database Server Out Of Use

MaxScale supports the concept of maintenance mode for servers within a cluster, this allows for planned, temporary removal of a database from the cluster within the need to change the MaxScale configuration.

To achieve the removal of a database server you can use the set server command in the maxadmin utility to set the maintenance mode flag for the server. This may be done interactively within maxadmin or by passing the command on the command line.

```
MaxScale> set server dbserver3 maintenance
MaxScale>
```

This will cause MaxScale to stop routing any new requests to the server, however if there are currently requests executing on the server these will not be interrupted.

w: www.mariadb.com

e: info@mariadb.com

To bring the server back into service use the “clear server” command to clear the maintenance mode bit for that server.

```
MaxScale> clear server dbserver3 maintenance
MaxScale>
```

Note that maintenance mode is not persistent, if MaxScale restarts when a node is in maintenance mode a new instance of MaxScale will not honour this mode. If multiple MaxScale instances are configured to use the node them maintenance mode must be set within each MaxScale instance. However if multiple services within one MaxScale instance are using the server then you only need set the maintenance mode once on the server for all services to take note of the mode change.