

By Falko Timme

Published: 2008-06-06 13:10

How To Repair MySQL Replication

Version 1.0

Author: Falko Timme <ft[at]falkotimme[dot]com>

Last edited 05/29/2008

If you have set up MySQL replication, you probably know this problem: sometimes there are invalid MySQL queries which cause the replication to not work anymore. In this short guide I explain how you can repair the replication on the MySQL slave without the need to set it up from scratch again.

I do not issue any guarantee that this will work for you!

1 Identifying The Problem

To find out whether replication is/is not working and what has caused to stop it, you can take a look at the logs. On Debian, for example, MySQL logs to `/var/log/syslog`:

```
grep mysql /var/log/syslog
```

```
server1:/home/admin# grep mysql /var/log/syslog
May 29 09:56:08 http2 mysqld[1380]: 080529 9:56:08 [ERROR] Slave: Error 'Table 'mydb.taggregate_temp_1212047760' doesn't exist' on query. Default database: 'mydb'. Query: 'UPDATE thread AS thread,taggregate_temp_1212047760 AS aggregate
May 29 09:56:08 http2 mysqld[1380]: ^ISET thread.views = thread.views + aggregate.views
May 29 09:56:08 http2 mysqld[1380]: ^IWHERE thread.threadid = aggregate.threadid', Error_code: 1146
May 29 09:56:08 http2 mysqld[1380]: 080529 9:56:08 [ERROR] Error running query, slave SQL thread aborted. Fix the problem, and restart the slave SQL thread with "SLAVE START". We stopped at log 'mysql-bin.001079' position 203015142
server1:/home/admin#
```

You can see what query caused the error, and at what log position the replication stopped.

To verify that the replication is really not working, log in to MySQL:

```
mysql -u root -p
```

On the MySQL shell, run:

```
mysql> SHOW SLAVE STATUS \G
```

If one of *Slave_IO_Running* or *Slave_SQL_Running* is set to *No*, then the replication is broken:

```
mysql> SHOW SLAVE STATUS G
```

```
***** 1. row *****
      Slave_IO_State: Waiting for master to send event
          Master_Host: 1.2.3.4
          Master_User: slave_user
          Master_Port: 3306
          Connect_Retry: 60
          Master_Log_File: mysql-bin.001079
      Read_Master_Log_Pos: 269214454
          Relay_Log_File: slave-relay.000130
          Relay_Log_Pos: 100125935
      Relay_Master_Log_File: mysql-bin.001079
      Slave_IO_Running: Yes
      Slave_SQL_Running: No
          Replicate_Do_DB: mydb
      Replicate_Ignore_DB:
          Replicate_Do_Table:
      Replicate_Ignore_Table:
      Replicate_Wild_Do_Table:
```

Replicate_Wild_Ignore_Table:

Last_Errno: 1146

Last_Error: Error 'Table 'mydb.taggregate_temp_1212047760' doesn't exist' on query. Default database: 'mydb'

.

Query: 'UPDATE thread AS thread,taggregate_temp_1212047760 AS aggregate

SET thread.views = thread.views + aggregate.views

WHERE thread.threadid = aggregate.threadid'

Skip_Counter: 0

Exec_Master_Log_Pos: 203015142

Relay_Log_Space: 166325247

Until_Condition: None

Until_Log_File:

Until_Log_Pos: 0

Master_SSL_Allowed: No

Master_SSL_CA_File:

Master_SSL_CA_Path:

Master_SSL_Cert:

Master_SSL_Cipher:

Master_SSL_Key:

Seconds_Behind_Master: NULL

1 row in set (0.00 sec)

mysql>

2 Repairing The Replication

Just to go sure, we stop the slave:

```
mysql> STOP SLAVE;
```

Fixing the problem is actually quite easy. We tell the slave to simply skip the invalid SQL query:

```
mysql> SET GLOBAL SQL_SLAVE_SKIP_COUNTER = 1;
```

This tells the slave to skip one query (which is the invalid one that caused the replication to stop). If you'd like to skip two queries, you'd use `SET GLOBAL SQL_SLAVE_SKIP_COUNTER = 2;` instead and so on.

That's it already. Now we can start the slave again...

```
mysql> START SLAVE;
```

... and check if replication is working again:

```
mysql> SHOW SLAVE STATUS \G
```

```
mysql> SHOW SLAVE STATUS G
***** 1. row *****
      Slave_IO_State: Waiting for master to send event
      Master_Host: 1.2.3.4
      Master_User: slave_user
      Master_Port: 3306
      Connect_Retry: 60
      Master_Log_File: mysql-bin.001079
      Read_Master_Log_Pos: 447560366
      Relay_Log_File: slave-relay.000130
      Relay_Log_Pos: 225644062
      Relay_Master_Log_File: mysql-bin.001079
      Slave_IO_Running: Yes
      Slave_SQL_Running: Yes
      Replicate_Do_DB: mydb
      Replicate_Ignore_DB:
      Replicate_Do_Table:
```

```
Replicate_Ignore_Table:
Replicate_Wild_Do_Table:
Replicate_Wild_Ignore_Table:
    Last_Errno: 0
    Last_Error:
    Skip_Counter: 0
Exec_Master_Log_Pos: 447560366
Relay_Log_Space: 225644062
Until_Condition: None
Until_Log_File:
Until_Log_Pos: 0
Master_SSL_Allowed: No
Master_SSL_CA_File:
Master_SSL_CA_Path:
Master_SSL_Cert:
Master_SSL_Cipher:
Master_SSL_Key:
Seconds_Behind_Master: 0
1 row in set (0.00 sec)
```

```
mysql>
```

As you see, both *Slave_IO_Running* and *Slave_SQL_Running* are set to *Yes* now.

Now leave the MySQL shell...

```
mysql> quit;
```

... and check the log again:

```
grep mysql /var/log/syslog
```

```
server1:/home/admin# grep mysql /var/log/syslog
May 29 09:56:08 http2 mysqld[1380]: 080529 9:56:08 [ERROR] Slave: Error 'Table 'mydb.taggregate_temp_1212047760' doesn't
exist' on query. Default database: 'mydb'. Query: 'UPDATE thread AS thread,taggregate_temp_1212047760 AS aggregate
May 29 09:56:08 http2 mysqld[1380]: ^ISET thread.views = thread.views + aggregate.views
May 29 09:56:08 http2 mysqld[1380]: ^IWHERE thread.threadid = aggregate.threadid', Error_code: 1146
May 29 09:56:08 http2 mysqld[1380]: 080529 9:56:08 [ERROR] Error running query, slave SQL thread aborted. Fix the problem,
and restart the slave SQL thread with "SLAVE START". We stopped at log 'mysql-bin.001079' position 203015142
May 29 11:42:13 http2 mysqld[1380]: 080529 11:42:13 [Note] Slave SQL thread initialized, starting replication in log
'mysql-bin.001079' at position 203015142, relay log '/var/lib/mysql/slave-relay.000130' position: 100125935
server1:/home/admin#
```

The last line says that replication has started again, and if you see no errors after that line, everything is ok.

3 Links

- MySQL: <http://www.mysql.com>