

Master/Slave Concept:

While hosting a website, the database is of utmost importance as a small glitch in the database can make or break the whole website. Hence, we look for different ways to keep the database secured. One of the common methods used is the MySQL master-slave replication.

The Master-slave data replication allows data to be replicated over multiple backups in-order to keep the data secured in case of any mishaps. Here, the master database is treated as an authoritative source and the slave databases are synchronized to it. The master logs the updates, which then ripple through to the slaves.

A database is “slaved” to a “master” when it receives a stream of updates from the master and database that receives functions as a copy.

Here, I’m going to mention a fairly unique issue where, a manual updation causes the replication to break and how this can be fixed.

Case history : In this case, there are two Database servers which works on the Master-Slave concept, which means one DB server is working as Master server and other as Slave server. Database syncing from Master to Slave is in such a way that the changes made in the Master server database (Update, Delete or Insert etc), would get reflected in the Slave server database.

Unfortunately, a manual updation was done in the databases of the Slave server and that lead to chaos. It was found that updates made in the Master was not getting reflected in the Slave. So, there lies a probability that the manual sync on the slave server had put the replication down or Slave goes out of sync with master. Hence, manual updation of the database breaks

Reset/Re-sync MySQL Master Slave Replication

Written by BiRU

Wednesday, 07 September 2022 21:31 -

the replication of database as the binary log file gets corrupted by the act. How to resolve this? Here is the solution!

This solution is based on some online research and this article will be a guide to reset MySQL replication.

Warning Message: *All the Binlog files will get deleted so necessary backup of the binlog files should be taken for future reference.*

In this case, 'test_replication' database is used in both the Master and the Slave server.

Before proceeding further, make sure to purge all the databases from the slave server and create the same in the slave server.

PHASE I : On Slave server

After login to the MySQL, perform below steps .

```
MariaDB [(none)]> DROP DATABASE test_replication;
```

```
MariaDB [(none)]> CREATE DATABASE test_replication;
```

First step is to stop slave on slave server and this can be achieved by executing the following command after logging in to MySQL

Reset/Re-sync MySQL Master Slave Replication

Written by BiRU

Wednesday, 07 September 2022 21:31 -

MariaDB [(none)]>> STOP SLAVE;

PHASE II :On Master Server

In this section, reset the Master state by executing the below given command after login into the MySQL.

MariaDB [(none)]>RESET MASTER;

MariaDB [(none)]>FLUSH TABLES WITH READ LOCK;

Lock Tables: MySQL enables client sessions to acquire table locks explicitly for the purpose of cooperating with other sessions to have access to the tables, or to prevent other sessions from modifying tables during periods, when a session requires exclusive access to them. A session can acquire or release locks only for itself. One session cannot acquire locks for another session or release locks held by another session. Locks may be used to emulate transactions or to get more speed when updating tables.

Now verify the status of the Master using the below command. The result of this command is as follows.

MariaDB [(none)]>SHOW MASTER STATUS;

Reset/Re-sync MySQL Master Slave Replication

Written by BiRU

Wednesday, 07 September 2022 21:31 -

```
MariaDB [(none)]> SHOW MASTER STATUS;
```

```
+-----+-----+-----+-----+
| File           | Position | Binlog_Do_DB | Binlog_Ignore_DB |
+-----+-----+-----+-----+
| mysql-bin.000001 |      312 |              |                  |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

~~When the replication is broken, you can reset the master to the original state. This is useful when you have a backup of the master and want to restore it to the original state.~~

```
MariaDB [(none)]> CHANGE MASTER TO MASTER_HOST='192.168.1.206', MASTER_G_POS=312;
```

```
Query OK, 0 rows affected (0.21 sec)
```

~~When the replication is broken, you can reset the master to the original state. This is useful when you have a backup of the master and want to restore it to the original state.~~

Reset/Re-sync MySQL Master Slave Replication

Written by BiRU

Wednesday, 07 September 2022 21:31 -

```
MariaDB [test_replication]> SHOW SLAVE STATUS\G;
***** 1. row *****
      Slave_IO_State: Waiting for master to send event
        Master_Host: 192.168.1.206
        Master_User: slave-01
        Master_Port: 3306
        Connect_Retry: 60
        Master_Log_File: mysql-bin.000001
    Read_Master_Log_Pos: 312
        Relay_Log_File: sanith1-relay-bin.000002
        Relay_Log_Pos: 535
    Relay_Master_Log_File: mysql-bin.000001
      Slave_IO_Running: Yes
      Slave_SQL_Running: Yes
        Replicate_Do_DB:
    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: 312
        Relay_Log_Space: 834
        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:
```