MySQL Master Master Repliction Tutorial

On this page

This tutorial describes how to set up MySQL master-master replication. We need to replicate MySQL servers to achieve high-availability (HA). In my case I need two masters that are synchronized with each other so that if one of them drops down, other could take over and no data is lost. Similarly when the first one goes up again, it will still be used as slave for the live one.

Here is a basic step by step tutorial, that will cover the mysql master and slave replication and also will describe the mysql master and master replication.

Notions: *we will call system 1 as master1 and slave2 and system2 as master2 and slave 1.*

Step 1:

Install mysql on master 1 and slave 1. configure network services on both system, like

Master 1/Slave 2 ip: 192.168.16.4

Master 2/Slave 1 ip : 192.168.16.5

Step 2:

On Master 1, make changes in my.cnf:

```
[mysqld]
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
old_passwords=1

log-bin
binlog-do-db=<database name>  # input the database which should be replicated
binlog-ignore-db=mysql            # input the database that should be ignored for replication
binlog-ignore-db=test

server-id=1

[mysql.server]
user=mysql
basedir=/var/lib


[mysqld_safe]
```

```
err-log=/var/log/mysqld.log
pid-file=/var/run/mysqld/mysqld.pid
```

Step 3:

On master 1, create a replication slave account in mysql.

```
mysql> grant replication slave on *.* to 'replication'@192.168.16.5 \
identified by 'slave';
```

and restart the mysql master1.

Step 4:

Now edit my.cnf on Slave1 or Master2 :

```
[mysqld]
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
old_passwords=1

server-id=2

master-host = 192.168.16.4
master-user = replication
master-password = slave
master-port = 3306

[mysql.server]
user=mysql
basedir=/var/lib

[mysqld_safe]
err-log=/var/log/mysqld.log
pid-file=/var/run/mysqld/mysqld.pid
```

Step 5:

Restart mysql slave 1 and at

```
mysql> start slave;
mysql> show slave status\G;
```

```
*************************** 1. row ***************************

           Slave_IO_State: Waiting for master to send event
              Master_Host: 192.168.16.4
              Master_User: replica
              Master_Port: 3306
            Connect_Retry: 60
          Master_Log_File: MASTERMYSQL01-bin.000009
      Read_Master_Log_Pos: 4
           Relay_Log_File: MASTERMYSQL02-relay-bin.000015
            Relay_Log_Pos: 3630
    Relay_Master_Log_File: MASTERMYSQL01-bin.000009
         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: 4
            Relay_Log_Space: 3630
            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: 1519187

  1 row in set (0.00 sec)
```

Above highlighted rows must be indicate related log files and Slave_IO_Running and   Slave_SQL_Running: must be to YES.

Step 6:

On master 1:

```
mysql> show master status;
+------------------------+----------+--------------+-------------------+
| File                   | Position | Binlog_Do_DB | Binlog_Ignore_DB |
+------------------------+----------+--------------+-------------------+
|MysqlMYSQL01-bin.000008 |      410 | adam         |                  |
+------------------------+----------+--------------+-------------------+
1 row in set (0.00 sec)
```

The above scenario is for master-slave, now we will create a slave master scenario for the same systems and it will work as master master.

Step 7:

On Master2/Slave 1, edit my.cnf and master entries into it:

```
 [mysqld]
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
# Default to using old password format for compatibility with mysql 3.x
# clients (those using the mysqlclient10 compatibility package).
old_passwords=1
server-id=2

master-host = 192.168.16.4
master-user = replication
master-password = slave
master-port = 3306

log-bin                     #information for becoming master added
binlog-do-db=adam

[mysql.server]
user=mysql
basedir=/var/lib
```

```
[mysqld_safe]
err-log=/var/log/mysqld.log
pid-file=/var/run/mysqld/mysqld.pid
```

Step 8:

Create a replication slave account on master2 for master1:

```
mysql> grant replication slave on *.* to 'replication'@192.168.16.4 identified by 'slave2';
```

Step 9:

Edit my.cnf on master1 for information of its master.

```
[mysqld]
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock

# Default to using old password format for compatibility with mysql 3.x
# clients (those using the mysqlclient10 compatibility package).
old_passwords=1


log-bin
binlog-do-db=adam
binlog-ignore-db=mysql
binlog-ignore-db=test

server-id=1
#information for becoming slave.
master-host = 192.168.16.5
master-user = replication
master-password = slave2
master-port = 3306


[mysql.server]user=mysqlbasedir=/var/lib
```

Step 10:

Restart both mysql master1 and master2.

On mysql master1:

```
mysql> start slave;
```

On mysql master2:

```
mysql > show master status;
```

On mysql master 1:

```
mysql> show slave status\G;
```

```
*************************** 1. row ***************************
             Slave_IO_State: Waiting for master to send event
                Master_Host: 192.168.16.5
                Master_User: replica
                Master_Port: 3306
              Connect_Retry: 60
            Master_Log_File: Mysql1MYSQL02-bin.000008
        Read_Master_Log_Pos: 410
             Relay_Log_File: Mysql1MYSQL01-relay-bin.000008
              Relay_Log_Pos: 445
      Relay_Master_Log_File: Mysql1MYSQL02-bin.000008
           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: 410
            Relay_Log_Space: 445
            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: 103799
1 row in set (0.00 sec)

ERROR:
No query specified
```

Check for the hightlighted rows, make sure its running. Now you can create tables in the database and you will see changes in slave. **Enjoy!!**

view as pdf | print

Share this page:

Suggested articles

44 Comment(s)

Add comment

Name *

Email *

B *I*

p

I'm not a robot

reCAPTCHA
Privacy - Terms

Submit comment

Comments

From: Chris Dean

Reply

Here are some additional notes on issues you might encounter during this process:
If your first Master DB (Master 1) contains views (at the time of writing) you'll not be able to use the command "load data from master" to clone Master 1 to Master 2 due to this bug: and you'll have to replicate the existing system some other way e.g. copy and paste data files or stream data via mysqldump

If you use auto incremeneting id's and you're using Master Master with both masters active you might want to consider setting the following options on Master 1 and Master 2 so that the auto increment values don't end up clashing in the case where an insert happens to occur at exactly the same time on both servers:

Make Master 1 only auto-increment odd numbers by adding this to my.cnf under [mysqld]:

```
auto_increment_increment= 2
auto_increment_offset   = 1
```

Make Master 2 only auto-increment even numbers by adding this to my.cnf under [mysqld]:
auto_increment_increment= 2
auto_increment_offset   = 2

This is particularly important if you're using auto increment id columns as primary keys

From: Anonymous Reply

for the autoincrement field if you run it buck wild on both u get data for a single field from both servers.
this breaks replication. error duplicate yadda yadda yadda
serverID 1
auto_increment_increment   2
auto_increment_offset   1
----
serverID 2
auto_increment_increment   2
auto_increment_offset   2
serverID 1 will start at 1 and increment 2 always giving odd numbers
serverID2 will start at 2 and increment by 2 always giving even numbers.
this will prevent the autoincrement field having duplicates when cross replicating.

From: Anonymous Reply

Do you think this setting is required for active-passive master-master replication?
I don't think as there will be no simulteneous writes that may require alternating auto increment requied!

From: Fred59 Reply

Thanks for this good howto.
Despite writing different auto_increment_offset in master 1 and master2, all my records are always with even number. How is it possible ?

From: Reply

One thing that is missing from the above, is how to deal with auto-increments so that you do not get collisions if you add data to both masters at the same time. There is an easy way to deal with this if you are using MySQL 5.0.2 or above.
In the my.cnf file on server1 add:
auto_increment_increment=2
auto_increment_offset=1
In the my.cnf file on server2 add:
auto_increment_increment=2
auto_increment_offset=2
This will make the auto-increment on server1 go, 1,3,5,7,9,etc... and on server2 go, 2,4,6,8,etc... thereby preventing collisions.
Andrew
www.andrewdixon.co.uk

From: Andreas Reply

How will this setup solve collisions in data changes?
"First wins?"
"Don't care?"

From: Matt Reply

It doesn't. If both nodes attempt to edit the same record at the same time, you could have problems. Not sure on the behavior, but most likely the last one to update would win.
The post described a way to deal with auto-increment data type, which is often use to uniquely index new records. The clear use of increment interval and offset allows new records to be created without collision.

From: Anonymous Reply

I read many comments about how good is using autoincrement to avoid collision.
No one cares about data? IDs are important for a lot of applications. Correlative IDs are expected usually. What would you think if you see a table an IDs are only even? "I'm loosing data!"

Since it's not the default behaviour for databases, it could create odd situations...

From: Ross Reply

I think it's a mistake to use id's for anything other than unique identification. I specifically think it's a mistake to attempt to extract business logic from an auto-generated id column. If you want an ordinal column that correlates to row creation time, add a creation_date column with a time value in it.

From: Iain Reply

Heh, +1 on that front. We had a system at work that used sequential ID numbers on entries (nothing to do with row IDs in a database but...), then we ran out of usable ID space, and switched to a random "try some numbers till you find a free one" (and later increased the usable ID space anyway). Turns out people were using it to count the number of new entries created every day and suddenly got nothing, then 1000x the normal volume due to high random numbers. Tsk tsk, told em how to do it *properly* (there's a created date field, they just weren't using it...)

From: Mohammad Khan Reply

Hi,

I am following the tutorial and trying to set up master master replication.In step 4, I added

master - host = <IP>

to .ini file on machine 2 (which is Master 2). The IP I am adding here is the IP of machine1(
which is master1).

However, when restart mysql server on machine 2, It does not start and following error
is shown in logs:

unknown variable master-host=<IP>.

Any suggestions to resolve this issue will be appreciated.

Thanks.

From: Anonymous Reply

There are breaking changes in Mysql 5.6 see:
http://dev.mysql.com/doc/refman/5.6/en/replication-options-slave.html

From: Will Fitch Reply

Agreed. InnoDB's row-level locking (or any locking for that matter) doesn't propagate via the replication protocol. This is a big disadvantage and the only solution is to implement a solution on the client code.

From: krex Reply

Try setting on both servers my.cnf:
auto_increment_increment=1
auto_increment_offset=2

From: Marco Fretz Reply

thank you for this great how-to. works pretty well for me.

From: Fabio Cecaro Reply

Hi sheikhsa,
In a Master-Master replication there are many problems with the autonumber field tables.
Because when a client insert a value in a table in a master on the other master we can have another client with another insert in the same table. So we have a possible conflict when the replication mechanism replies these two records, because if this ID is relationate with another table I think we corrupting the relational logical.
We must fix addend two lines in the my.cnf of two masters

auto_increment_increment
auto_increment_offset
Example in this case we can set to the first Master the values:
auto_increment_increment = 2
auto_increment_offset = 1
and in the other:
auto_increment_increment = 2
auto_increment_offset = 2.
So in one master we generate only the pairs number and dispair in the other, when the replication come we haven't conflicts.

From: tom Reply

hi adam,
thanks for putting this up. just one possible typo:
in step 2 you have
....
old_passwords=1

log-bin
binlog-do-db=<database name> # input the database which should be replicated
....
shouldn't log-bin be followed with a filename like ?
log-bin = /var/log/mysql/mysql-bin.log

cheers,
tom

From: Anonymous Reply

Dear, I have done the Multi master Replication with the proceedure described above. No records are being replicated on both servers. and the status of the Slave shows: Slave_IO_Running: NO Slave_SQL_Running: Yes And same thing are shown on both Servers when looking at the 'show slave status\G;' Please help me out how i can rectify this problem. Regards, Majid

> **From:** Lars Reply
>
> Please make sure that you have given each server a uniqe server-id

From: Glen Bremner-Stokes Reply

Thank you for a great post and for taking the time to put it out there.
I now have my sugarcrm system on the web and a local sugarcrm on my laptop working bi-directionally. So I can update a record in either system and the other is instantly updated.
Very very cool and really easy following your instructions. Well done.

From: Reply

Today , While trying this article, I was keep getting this error for Step 9(Changing Master1 as Slaves for Master2)

091123 18:59:13 [ERROR] Slave I/O: Fatal error: The slave I/O thread stops because master and slave have equal MySQL server ids; these ids must be different for replication to work (or the --replicate-same-server-id
option must be used on slave but this does not always make sense; please check the manual before using it). Error_code: 1593

I don't know if I was doing anything wrong or not.
but by adding "log-slave-updates" solved my issue.
so it was like this :
server-id=1
#information for becoming slave.
master-host = 192.168.16.5
master-user = replication
master-password = slave2
master-port = 3306

log-slave-updates

Ref :
(a) http://dev.mysql.com/doc/refman/5.0/en/replication-options-slave.html#option_mysqld_replicate-same-server-id

(b) http://dev.mysql.com/doc/refman/5.0/en/replication-options-slave.html#option_mysqld_log-slave-updates
Thanks

From: Hassan Reply

do you have any steps for us to setup load balancer with master-master replication what you described above?
if we are using PHP5, where does the connection point to in the master-master replication?
I'm looking for failover in this setup, how does it work?
regards

From: grozours Reply

I suppose you have the same mysql username and password for both databases.
I suppose you have the same replication username and password for both databases.

An environment script mysql-env.sh :

export mysql_username=**your_mysql_user**
export mysql_password=**your_mysql_user_password**
export master1=**hostname_of_master1**
export master2=**hostname_of_master2**
export master1_ip=**ip_of_master1**
export master2_ip=**ip_of_master2**
export replication_user=**your_replication_user**
export replication_password=**your_replication_password**

Check mysql replication status script check-mysql-replication.sh:

```
#!/bin/bash
source mysql-env.sh

Master1_Slave_IO_Running=$(mysql -Bse "show slave status\G" -u ${mysql_username} -p${mysql_password} -h $master1 | grep Slave_IO_Running | awk '{ print $2 }';)
Master1_Slave_SQL_Running=$(mysql -Bse "show slave status\G" -u ${mysql_username} -p${mysql_password} -h $master1 | grep Slave_SQL_Running | awk '{ print $2 }';)
Master1_Slave_Master_Log_File=$(mysql -Bse "show slave status\G" -u ${mysql_username} -p${mysql_password} -h $master1 | grep "\ Master_Log_File" | awk '{ print $2 }';)
Master1_Slave_Read_Master_Log_Pos=$(mysql -Bse "show slave status\G" -u ${mysql_username} -p${mysql_password} -h $master1 | grep Read_Master_Log_Pos | awk '{ print $2 }';)

Master2_Slave_IO_Running=$(mysql -Bse "show slave status\G" -u ${mysql_username} -p${mysql_password} -h $master2 | grep Slave_IO_Running | awk '{ print $2 }';)
Master2_Slave_SQL_Running=$(mysql -Bse "show slave status\G" -u ${mysql_username} -p${mysql_password} -h $master2 | grep Slave_SQL_Running | awk '{ print $2 }';)
Master2_Slave_Master_Log_File=$(mysql -Bse "show slave status\G" -u ${mysql_username} -p${mysql_password} -h $master2 | grep "\ Master_Log_File" | awk '{ print $2 }';)
Master2_Slave_Read_Master_Log_Pos=$(mysql -Bse "show slave status\G" -u ${mysql_username} -p${mysql_password} -h $master2 | grep Read_Master_Log_Pos | awk '{ print $2 }';)

Master1_Master_File=$(mysql -Bse "show master status\G" -u ${mysql_username} -p${mysql_password} -h $master1 | grep File | awk '{ print $2 }';)
Master1_Master_Position=$(mysql -Bse "show master status\G" -u ${mysql_username} -p${mysql_password} -h $master1 | grep Position | awk '{ print $2 }';)

Master2_Master_File=$(mysql -Bse "show master status\G" -u ${mysql_username} -p${mysql_password} -h $master2 | grep File | awk '{ print $2 }';)
Master2_Master_Position=$(mysql -Bse "show master status\G" -u ${mysql_username} -p${mysql_password} -h $master2 | grep Position | awk '{ print $2 }';)

# Master 1($master1) / Slave 1($master2)
```

```
echo "";
echo "------------------------------------------";
echo "Master 1($master1) / Slave 1($master2) Status :";
echo "------------------------------------------";
echo "Master : $Master1_Master_File - Position : $Master1_Master_Position"
echo "Slave  : $Master2_Slave_Master_Log_File - Position : $Master2_Slave_Read_Master_Log_Pos"
echo "Slave_IO_Running  : $Master1_Slave_IO_Running"
echo "Slave_SQL_Running : $Master1_Slave_SQL_Running"
echo "";
if [ "$Master1_Master_File" == "$Master2_Slave_Master_Log_File" ] && [ "$Master1_Master_Position" ==
"$Master2_Slave_Read_Master_Log_Pos" ] && [ "$Master1_Slave_IO_Running" == "Yes" ] && [ "$Master1_Slave_SQL_Running" ==
"Yes" ];
then
echo "Status OK";
else
echo "Status KO";
fi
echo "";


# Master 2($master2) / Slave 2($master1)

echo "------------------------------------------";
echo "Master 2($master2) / Slave 2($master1)"
echo "------------------------------------------";
echo "Master : $Master2_Master_File - Position : $Master2_Master_Position"
echo "Slave  : $Master1_Slave_Master_Log_File - Position : $Master1_Slave_Read_Master_Log_Pos"
echo "Slave_IO_Running  : $Master2_Slave_IO_Running"
echo "Slave_SQL_Running : $Master2_Slave_SQL_Running"
echo "";
if [ "$Master2_Master_File" == "$Master1_Slave_Master_Log_File" ] && [ "$Master2_Master_Position" ==
"$Master1_Slave_Read_Master_Log_Pos" ] && [ "$Master2_Slave_IO_Running" == "Yes" ] && [ "$Master2_Slave_SQL_Running" ==
"Yes" ];
then
        echo "Status OK";
else
        echo "Status KO";
fi
echo "";
```

Restore mysql replication script restore-mysql-replication.sh:

```
 #!/bin/bash
source mysql-env.sh

ps -ef | grep -v grep | grep -q  mysql-$master1
if [ "$?" -eq "1" ];
then
screen -dmS mysql-$master1 mysql -u ${mysql_username} -p${mysql_password} -h $master1
fi

ps -ef | grep -v grep | grep -q  mysql-$master2
if [ "$?" -eq "1" ];
then
screen -dmS mysql-$master2 mysql -u ${mysql_username} -p${mysql_password} -h $master2
fi

screen -S mysql-$master1 -X -p0 eval "stuff 'STOP SLAVE;'^M"
screen -S mysql-$master1 -X -p0 eval "stuff 'FLUSH TABLES WITH READ LOCK;'^M"

screen -S mysql-$master2 -X -p0 eval "stuff 'STOP SLAVE;'^M"
screen -S mysql-$master2 -X -p0 eval "stuff 'FLUSH TABLES WITH READ LOCK;'^M"
```

```
Master2_Master_File=$(mysql -Bse "show master status\G" -u ${mysql_username} -p${mysql_password} -h $master2 | grep File | awk '{ print $2 }';)
Master2_Master_Position=$(mysql -Bse "show master status\G" -u ${mysql_username} -p${mysql_password} -h $master2 | grep Position | awk '{ print $2 }';)

screen -S mysql-$master1 -X -p0 eval "stuff 'CHANGE MASTER TO MASTER_HOST="'"'${master2_ip}'"'", MASTER_PORT=3306,MASTER_USER="'"'${replication_user}'"'", MASTER_PASSWORD="'"'${replication_password}'"'";^M"
screen -S mysql-$master2 -X -p0 eval "stuff 'CHANGE MASTER TO MASTER_HOST="'"'${master1_ip}'"'", MASTER_PORT=3306,MASTER_USER="'"'${replication_user}'"'", MASTER_PASSWORD="'"'${replication_password}'"'";^M"


screen -S mysql-$master1 -X -p0 eval "stuff 'CHANGE MASTER TO MASTER_LOG_FILE="'"'$Master2_Master_File'"'",MASTER_LOG_POS=$Master2_Master_Position;'^M"

Master1_Master_File=$(mysql -Bse "show master status\G" -u ${mysql_username} -p${mysql_password} -h $master1 | grep File | awk '{ print $2 }';)
Master1_Master_Position=$(mysql -Bse "show master status\G" -u ${mysql_username} -p${mysql_password} -h $master1 | grep Position | awk '{ print $2 }';)

screen -S mysql-$master2 -X -p0 eval "stuff 'CHANGE MASTER TO MASTER_LOG_FILE="'"'$Master1_Master_File'"'",MASTER_LOG_POS=$Master1_Master_Position;'^M"

screen -S mysql-$master2 -X -p0 eval "stuff 'START SLAVE;'^M"
screen -S mysql-$master2 -X -p0 eval "stuff 'UNLOCK TABLES;'^M"

screen -S mysql-$master1 -X -p0 eval "stuff 'START SLAVE;'^M"
screen -S mysql-$master1 -X -p0 eval "stuff 'UNLOCK TABLES;'^M"

screen -S mysql-$master2 -X -p0 eval "stuff 'quit;'^M"
screen -S mysql-$master1 -X -p0 eval "stuff 'quit;'^M"
```

From: Charles-Alban Allard Reply

Thank you for those scripts.
On the first script, there is a little error, I think.

The end shoul be:

```
# Master 1($master1) / Slave 1($master2)


echo "";

echo "--------------------------------------------";

echo "Master 1($master1) / Slave 1($master2) Status :";

echo "--------------------------------------------";

echo "Master : $Master1_Master_File - Position : $Master1_Master_Position"

echo "Slave  : $Master2_Slave_Master_Log_File - Position : $Master2_Slave_Read_Master_Log_Pos"

echo "Slave_IO_Running  : $Master2_Slave_IO_Running"

echo "Slave_SQL_Running : $Master2_Slave_SQL_Running"

echo "";

if [ "$Master2_Master_File" == "$Master1_Slave_Master_Log_File" ] && [ "$Master2_Master_Position" == "$Master1_Slave_Read_Master_Log_Pos" ] && [ "$Master2_Slave_IO_Running" == "Yes" ] && [ "$Master2_Slave_SQL_Running" == "Yes" ];
```

```
then
        echo "Status OK";
else
        echo "Status KO";
fi
echo "";

# Master 2($master2) / Slave 2($master1)

echo "-------------------------------------------";
echo "Master 2($master2) / Slave 2($master1)"
echo "-------------------------------------------";
echo "Master : $Master2_Master_File - Position : $Master2_Master_Position"
echo "Slave  : $Master1_Slave_Master_Log_File - Position : $Master1_Slave_Read_Master_Log_Pos"
echo "Slave_IO_Running  : $Master1_Slave_IO_Running"
echo "Slave_SQL_Running : $Master1_Slave_SQL_Running"
echo "";
if [ "$Master1_Master_File" == "$Master2_Slave_Master_Log_File" ] && [ "$Master1_Master_Position" == "$Master2_Slave_Read_Master_Log_Pos" ] && [ "$Master1_Slave_IO_Running" == "Yes" ] && [ "$Master1_Slave_SQL_Running" == "Yes" ];
then
echo "Status OK";
else
echo "Status KO";
fi
echo "";
```

From: Nico Reply

Yes, but the final check should be:

```
# Master 1($master1) / Slave 1($master2)
if [ "$Master1_Master_File" == "$Master2_Slave_Master_Log_File" ] && [ "$Master1_Master_Position" == "$Master2_Slave_Read_Master_Log_Pos" ] && [ "$Master2_Slave_IO_Running" == "Yes" ] && [ "$Master2_Slave_SQL_Running" == "Yes" ];

# Master 2($master2) / Slave 2($master1)
if [ "$Master2_Master_File" == "$Master1_Slave_Master_Log_File" ] && [ "$Master2_Master_Position" == "$Master1_Slave_Read_Master_Log_Pos" ] && [ "$Master1_Slave_IO_Running" == "Yes" ] && [ "$Master1_Slave_SQL_Running" == "Yes" ];
```

From: thomas Reply

Sorry, i am really new to this.

where do i place
mysql-env.sh
check-mysql-replication.sh
restore-mysql-replication.sh

From: Eggert Reply

While this is a great tutorial, there seems to be a glitch in step 2:
binlog-do-db=<database name> # input the database which should be replicated
should read:
replicate-do-db=<database name> # input the database which should be replicated
Agree?

From: Michael Pawlowsky Reply

From the mysql.com site:
The server evaluates the `--binlog-do-db` and `--binlog-ignore-db` options in the same way as it does the `--replicate-do-db` and `--replicate-ignore-db` options. For information about how this is done, seeSection 15.2.3.1, "Evaluation of Database-Level Replication and Binary Logging Options".

From: Luis Loriente Reply

Great post, I have applied this configuration and works fine, but I think there is a problem if we want a point-in-time recovery for data-loss or accidental corruption of database.
To get this in an usual configuration (just one mysql server) we have to start from a snapshot with mysqldump (or other) and then apply the log_bin of the server to the point we want, or we can manipulate this log-bin to extract undesired lines.
With master-master configuration we have two log-bin groups (one from each server) to apply to the initial mysqldump snapshot, so we would need some kind of tool to merge these two log_bin groups in chronological order to get just one and apply it to the initial snapshot.
Due to the master-master configuration we can't activate the log_slave_updates which would generate a full log_bin of changes.
Has anyone thought about this?

From: Adam Reply

Hi,
At the end of the session you're getting the "ERROR: No query specified" because your "show slave status" ends with "\G;".
The "\G" token ends a query, so following that you have zero bytes and then another end query token, same as if you'd done "show slave status;;"

From: Anonymous Reply

ERROR 1200 (HY000): The server is not configured as slave; fix in config file or with CHANGE MASTER TO
pls help me..............

From: Anonymous Reply

try "FLUSH SLAVE", then do your "CHANGE MASTER TO ..." and then "START SLAVE" again, this worked for me,

Hopefully it works for you too.

From: Ramanath Reply

Master2/Slave1 is showing both slave staus and master status.but master1/slave2 is showing only master status and if i given start slave command then the below error is coming.So please help me
ERROR 1200 (HY000): The server is not configured as slave; fix in config file or with CHANGE MASTER TO

From: Aamir Hussain Reply

Hi, its really a great tutorial can anybody tell me how i can check High Availability in case of failover?

From: Al Reply

a step by step for creating replication with examples is listed at How to create Replication on Mysql: Step by step

From: Sushil Jain Reply

Thanks for the script to check DB replication and restore DB replication. However restore DB replication script will end up into further trouble since it will make both the DB out of sync in terms of data which can break the replication at later point of time. If restore script can include taking DB dump from one DB and restoring it to another DB and then do all the settings, that would be a real restore.

From: Duncan Reply

Hi, i tried with step 4, but I have this problem:
sudo /etc/init.d/mysqld start
MySQL Daemon failed to start.
Starting mysqld: [FAILED]
Can I help me please?
Bye

From: Douglas Reply

In a Master-Master Replication, If one server goes down for a long period. Whats happen when this server goes up again? The replication system will automatic sync the server without any manual intervention ?

From: till Reply

Yes, as long as the outage is not longer than the blinlig storage time.

From: Douglas Reply

Thank you for your reply!

From: Ashaduzzaman Bhuiyan Reply

Dear Sir ,
I did both way communication by replication on PC1 and PC2 its work fine . Problem is Another pc PC3 is only slave and i set its master is PC1 . When any change Occure in PC1 its hits Both PC2 and PC3 .But When any change occure in PC2 its Only Occure in PC2 , PC3 Doesn't receive any change whats problem .
NOTE . PC1 and PC2 Both are Master and salve ,   PC3 Only Slave Connected PC1 as master

From: Lars Behrens Reply

Hey everyone. I'd thought i share my automated resync mariadb/mysql script.
It’s possible that the master-master (or master-slave) setup get out of sync. This can happen when, for example, you set back snapshot, or in some really exceptional cases, when two queries are send at the same time to different databases with the same database/table as destination. When a out-of-sync occurs, without you knowing it, it’ll cause a big problem. because from that point on, the servers won’t be synchronising data anymore. And because we have a load-balancer with multiple databases, people will keep sending and storing data in different databases. if you find out, after lets say a week, that you’re databases aren’t in sync anymore, than you’ll have to drop a week of data for one database. To overcome this problem, i wrote a script that checks if the database is in sync every 10 minutes, and if not, automatically synchronizes with the master database.
[code]

```
#First we grab the master status of server 2(the other server) and the slave state (of this server)
#-----------------------------------------------------------------
#MAKE SURE YOU EDIT THE *MASTERSERVERIP* TO THE IP OF YOUR SECOND SERVER AND *PASSWORD* TO YOUR
MYSQL PASSWORD
MASTER_STATUS=$(ssh root@*MASTERSERVERIP* 'mysql -uroot -p*PASSWORD* -e"SHOW MASTER STATUS;";' | awk '{$1=$1}
NR==2')

SLAVE_STATUS=$(mysql -uroot -p'*PASSWORD*' -e "SHOW SLAVE STATUS\G" | awk '{$1=$1} NR==2 || NR==7 || NR==8')

#-----------------------------------------------------------------

#The variable MASTER_STATUS contains the MASTER_LOG_FILE and the MASTER_LOG_POS, if a out-of-sync occurs we want to
provide our slave with these values.
#Since MASTER_STATUS is now containing exactly for example: "mariadb-bin.000073 317858042"
#We'll make a new variable from this variable, so we have two clean variables containing the log file and position. We'll use these
later as variable in our query to update the sync status of the database.
# the awk '{print $1;} makes sure that we only get the first word (mariadb-bin.000073) and awk '{print $2;} the second word
(317858042)
```

```
MASTER_LOG_FILE=$(echo "$MASTER_STATUS" | awk '{print $1;}')

MASTER_LOG_POS=$(echo "$MASTER_STATUS" | awk '{print $2;}')
#-----------------------------------------------------------------

#For our comparisation (the check if everything is running fine) we need to check the SLAVE_IO_STATE.
#SLAVE_STATUS contains the following 3 lines at the moment:
#
#Slave_IO_State: Waiting for master to send event
#Master_Log_File: mariadb-bin.000073
#Read_Master_Log_Pos: 317858042
#
#We want to extract each part after the ": " part. So we can compare the string "Waiting for master to send event". This can be done with: "awk 'NR==1' | sed -n -e 's/^.*: //p'".
#We use awk to get a certain line number and sed to print the part after a certain pattern, in this case ": "
#If the string contains something else, it means that the database is out of sync.
#The slave master log file and log position can be printed afterward to make sure that the master and slave are in sync now.

SLAVE_IO_STATE=$(echo "$SLAVE_STATUS" | awk 'NR==1' | sed -n -e 's/^.*: //p')
#-----------------------------------------------------------------
#We don't need these variables. Altough they could come in handy for later (for some reason, don't know what yet?) for now i'll coment these two out
SLAVE_LOG_FILE=$(echo "$SLAVE_STATUS" | awk 'NR==2' | sed -n -e 's/^.*: //p')
SLAVE_LOG_POS=$(echo "$SLAVE_STATUS" | awk 'NR==3' | sed -n -e 's/^.*: //p')
#-----------------------------------------------------------------
printf "\n\n\n\n"
#-----------------------------------------------------------------

#We now have all the information needed to check if the server is still in sync, and if not, sending a query with the right log pos and file.

if [ "$SLAVE_IO_STATE" != "Waiting for master to send event" ]

then

printf "\n\nTHE SLAVE STATE IS: \n"
echo "$SLAVE_IO_STATE"

printf "\n\nTHE MASTER LOG FILE IS: \n"
echo "$MASTER_LOG_FILE"

printf "\n\nTHE SLAVE LOG FILE IS: \n"
echo "$SLAVE_LOG_FILE"

printf "\n\nTHE MASTER LOG POSITION IS: \n"
echo "$MASTER_LOG_POS"

printf "\n\nTHE SLAVE LOG POSITION IS: \n"
echo "$SLAVE_LOG_POS"
printf "\n\n\nSERVER IS OUT OF SYNC \nRESYNCING SERVER NOW!!! \nPLEASE  WAIT!!!\n\n\n"

mysql -uroot -p'*PASSWORD*' -e "STOP SLAVE;"
mysql -uroot -p'*PASSWORD*' -e "CHANGE MASTER TO MASTER_LOG_FILE='$MASTER_LOG_FILE', MASTER_LOG_POS=$MASTER_LOG_POS;"
mysql -uroot -p'*PASSWORD*' -e "START SLAVE;"
MASTER_STATUS=$(ssh root@*MASTERSERVERIP* 'mysql -uroot -p*PASSWORD* -e"SHOW MASTER STATUS;";' | awk '{$1=$1} NR==2')
SLAVE_STATUS=$(mysql -uroot -p'*PASSWORD*' -e "SHOW SLAVE STATUS\G" | awk '{$1=$1} NR==2 || NR==7 || NR==8')
MASTER_LOG_FILE=$(echo "$MASTER_STATUS" | awk '{print $1;}')
MASTER_LOG_POS=$(echo "$MASTER_STATUS" | awk '{print $2;}')
SLAVE_IO_STATE=$(echo "$SLAVE_STATUS" | awk 'NR==1' | sed -n -e 's/^.*: //p')
SLAVE_LOG_FILE=$(echo "$SLAVE_STATUS" | awk 'NR==2' | sed -n -e 's/^.*: //p')
SLAVE_LOG_POS=$(echo "$SLAVE_STATUS" | awk 'NR==3' | sed -n -e 's/^.*: //p')

printf "\n\nTHE SLAVE STATE IS: \n"
echo "$SLAVE_IO_STATE"

printf "\n\nTHE MASTER LOG FILE IS: \n"
echo "$MASTER_LOG_FILE"
```

```
printf "\n\nTHE SLAVE LOG FILE IS: \n"
echo "$SLAVE_LOG_FILE"

printf "\n\nTHE MASTER LOG POSITION IS: \n"
echo "$MASTER_LOG_POS"

printf "\n\nTHE SLAVE LOG POSITION IS: \n"
echo "$SLAVE_LOG_POS"

else

echo "SERVER IS IN SYNC"
printf "\n\nTHE SLAVE STATE IS: \n"
echo "$SLAVE_IO_STATE"

printf "\n\nTHE MASTER LOG FILE IS: \n"
echo "$MASTER_LOG_FILE"

printf "\n\nTHE SLAVE LOG FILE IS: \n"
echo "$SLAVE_LOG_FILE"

printf "\n\nTHE MASTER LOG POSITION IS: \n"
echo "$MASTER_LOG_POS"

printf "\n\nTHE SLAVE LOG POSITION IS: \n"
echo "$SLAVE_LOG_POS"
fi
printf "\n\n\n\n"
[/code]
```

From: Anyone Reply

Is this is required to perform databse equalization, if i have existing server with diffrent no. of database and different amount of data in both server? Before starting the master to master replication. While i want to replicate only some of them database.

From: Betty Reply

Hello.. What happen if sometimes slave is offline? Will replication continue working when slave become online again? Thanks for explanation.

Tutorials › **MySQL Master Master Replication**

Sign up now!

Tutorial Info

Author:	sheikhsa
Tags:	mysql

Share This Page

Recommend Tweet Follow G+

40.2k Followers

Popular Tutorials

Ubuntu 18.04 (LTS) LAMP server tutorial with Apache, PHP 7.2, and MySQL

How to use grep to search for strings in files on the shell

Setting, Changing And Resetting MySQL Root Passwords

How to Monitor Nginx using Netdata on CentOS 7

Debian 9 LAMP Server Tutorial with Apache, PHP 7 and MariaDB

The Perfect Server - Ubuntu 16.04 (Xenial Xerus) with Apache, PHP, MySQL, PureFTPD, BIND, Postfix, Dovecot and ISPConfig 3.1

How to Install CodeIgniter PHP Framework on Ubuntu 18.04 LTS

How to use the Linux ftp command to up- and download files on the shell

Linux Basics - Set a Static IP on Ubuntu

How to Install Subsonic Media Server Ubuntu 18.04

Xenforo skin by Xenfocus

Contribute Contact Help Imprint

Terms