How to re-sync the Mysql DB if Master and slave have different database incase of Mysql replication?

Asked 13 years, 1 month ago Modified 12 days ago Viewed 290k times

159

Mysql `Server1` is running as **MASTER**.
Mysql `Server2` is running as **SLAVE**.

Now DB replication is happening from **MASTER** to **SLAVE**.

`Server2` is removed from network and re-connect it back after 1 day. After this there is mismatch in database in master and slave.

How to re-sync the DB again as after restoring DB taken from Master to Slave also doesn't solve the problem ?

mysql database database-replication

Share Improve this question Follow

edited Feb 26, 2013 at 17:19

Shamoon
40.5k 88 298 553

asked Mar 2, 2010 at 19:20

Indu Sharma
1,591 3 10 5

13 Answers

Sorted by: Highest score (default)

322

This is the full step-by-step procedure to resync a master-slave replication from scratch:

At the master:

```
RESET MASTER;
FLUSH TABLES WITH READ LOCK;
SHOW MASTER STATUS;
```

And **copy the values of the result** of the last command somewhere.

Without closing the connection to the client (because it would release the read lock) issue the command to get a dump of the master:

```
mysqldump -u root -p --all-databases > /a/path/mysqldump.sql
```

Now you can release the lock, even if the dump hasn't ended yet. To do it, perform the following command in the MySQL client:

```
UNLOCK TABLES;
```

Now copy the dump file to the slave using scp or your preferred tool.

At the slave:

Open a connection to mysql and type:

```
STOP SLAVE;
```

Load master's data dump with this console command:

```
mysql -uroot -p < mysqldump.sql
```

Sync slave and master logs:

```
RESET SLAVE;
CHANGE MASTER TO MASTER_LOG_FILE='mysql-bin.000001', MASTER_LOG_POS=98;
```

Where the values of the above fields are the ones you copied before.

Finally, type:

```
START SLAVE;
```

To check that everything is working again, after typing:

```
SHOW SLAVE STATUS;
```

you should see:

```
Slave_IO_Running: Yes
Slave_SQL_Running: Yes
```

That's it!

edited Aug 15, 2016 at 10:12

vauhochzett

2,404 2 17 37

answered Jul 12, 2010 at 15:08

David Espart

11.4k 7 36 50

8 With INNODB typed dabatase and other complex column types like BLOB and DATE defined, I recommend using the following switches: `--opt --single-transaction --comments --hex-blob --dump-date --no-autocommit --all-databases` – Ken Pega May 31, 2011 at 2:21

5 Is `RESET_SLAVE` necessary? Note that these instructions reset the replication user and password, so you'll have to reenter those with `CHANGE MASTER TO...` – Mike S. May 31, 2012 at 15:35

32 If you use the --master-data flag when calling mysqldump on the master, the CHANGE MASTER TO command is written into the dump file and thus saves the step of executing it after importing the dump file into the slave. – udog Apr 14, 2013 at 23:04

3 Not locking the master (doesn't require Percona) plusbryan.com/mysql-replication-without-downtime Another benefit of this is the SQL dump also comes with the necessary "CHANGE MASTER" line (commented out) – mahemoff Apr 19, 2013 at 6:17

2 Is there a way to automate this? – Metafaniel Jan 5, 2015 at 20:49

36

The documentation for this at the MySQL site is woefully out of date and riddled with foot-guns (such as interactive_timeout). Issuing FLUSH TABLES WITH READ LOCK as part of your export of the master generally only makes sense when coordinated with a storage/filesystem snapshot such as LVM or zfs.

If you are going to use mysqldump, you should rely instead on the --master-data option to guard against human error and release the locks on the master as quickly as possible.

Assume the master is 192.168.100.50 and the slave is 192.168.100.51, each server has a distinct server-id configured, the master has binary logging on and the slave has read-only=1 in my.cnf

To stage the slave to be able to start replication just after importing the dump, issue a CHANGE MASTER command but omit the log file name and position:

```
slaveserver> CHANGE MASTER TO MASTER_HOST='192.168.100.50',
MASTER_USER='replica', MASTER_PASSWORD='asdmk3qwdq1';
```

Issue the GRANT on the master for the slave to use:

```
masterserver> GRANT REPLICATION SLAVE ON *.* TO 'replica'@'192.168.100.51'
IDENTIFIED BY 'asdmk3qwdq1';
```

Export the master (in screen) using compression and automatically capturing the correct binary log coordinates:

```
mysqldump --master-data --all-databases --flush-privileges | gzip -1 >
replication.sql.gz
```

Copy the replication.sql.gz file to the slave and then import it with zcat to the instance of MySQL running on the slave:

```
zcat replication.sql.gz | mysql
```

Start replication by issuing the command to the slave:

```
slaveserver> START SLAVE;
```

Optionally update the /root/.my.cnf on the slave to store the same root password as the master.

If you are on 5.1+, it is best to first set the master's binlog_format to MIXED or ROW. Beware that row logged events are slow for tables which lack a primary key. This is usually better than the alternative (and default) configuration of binlog_format=statement (on master), since it is less likely to produce the wrong data on the slave.

If you must (but probably shouldn't) filter replication, do so with slave options replicate-wild-do-table=dbname.% or replicate-wild-ignore-table=badDB.% and use only binlog_format=row

This process will hold a global lock on the master for the duration of the mysqldump command but will not otherwise impact the master.

If you are tempted to use mysqldump --master-data --all-databases --single-transaction (because you only using InnoDB tables), you are perhaps better served using MySQL Enterprise Backup or the open source implementation called xtrabackup (courtesy of Percona)

Share Improve this answer Follow

edited Oct 29, 2013 at 18:14

answered Oct 29, 2013 at 17:22

Outdated
361 3 3

3 If you want to simply rebuild an existing slave, you can follow the above process, skipping a couple of steps: The GRANT and manual CHANGE MASTER command – Outdated Oct 29, 2013 at 17:25

Question 1: On windows, what would be an equivalent of `zcat` . Question 2: How does this `mysqldump` fare with large databases in terms of performance? Any way to use `SELECT INTO OUTFILE` and `LOAD DATA` smoothly in your suggested process? (Since they generally perform faster) – Ifedi Okonkwo Jul 12, 2016 at 19:35

No need to `STOP SLAVE` somewhere in the process ? – David V. Apr 21, 2020 at 8:47

20

Unless you are writing directly to the slave (Server2) the only problem should be that Server2 is missing any updates that have happened since it was disconnected. Simply restarting the slave with "START SLAVE;" should get everything back up to speed.

Share Improve this answer Follow

answered Mar 8, 2010 at 22:14

malonso
2,247 1 21 33

4 You can check the bin logs and see if they cover the period in time your system was out of sync. If you are missing days this might be a bigger problem and require you to do a full dump to restore missing data. – nelaaro May 17, 2017 at 16:19

10

I am very late to this question, however I did encounter this problem and, after much searching, I found this information from Bryan Kennedy: http://plusbryan.com/mysql-replication-without-downtime

On Master take a backup like this:
mysqldump --skip-lock-tables --single-transaction --flush-logs --hex-blob --master-data=2 -A > ~/dump.sql

Now, examine the head of the file and jot down the values for MASTER_LOG_FILE and MASTER_LOG_POS. You will need them later: **head dump.sql -n80 | grep "MASTER_LOG"**

Copy the "dump.sql" file over to Slave and restore it: **mysql -u mysql-user -p < ~/dump.sql**

Connect to Slave mysql and run a command like this: **CHANGE MASTER TO MASTER_HOST='master-server-ip', MASTER_USER='replication-user', MASTER_PASSWORD='slave-server-password', MASTER_LOG_FILE='value from above', MASTER_LOG_POS=value from above; START SLAVE;**

To check the progress of Slave: **SHOW SLAVE STATUS;**

If all is well, Last_Error will be blank, and Slave_IO_State will report “Waiting for master to send event”. Look for Seconds_Behind_Master which indicates how far behind it is. YMMV. :)

Share Improve this answer Follow

answered Jan 20, 2016 at 19:13
Jeffery7
101 1 3

4 This works, and if the slave already is set up and only fell out of sync, you need not run CHANGE MASTER; just specify `--master-data=1` (or just `--master-data`). – LSerni Oct 13, 2016 at 9:06

1 Whoa, that was amazing, it worked brilliantly! And indeed, using just `--master-data` puts the `CHANGE MASTER` command as an executable query, rather than a comment, so you just import the file, run `SLAVE START` and that's it. Thanks a lot! – LachoTomov Oct 7, 2022 at 12:34

8

I think, Maatkit utilits helps for you! You can use mk-table-sync. Please see this link:
http://www.maatkit.org/doc/mk-table-sync.html

Share Improve this answer Follow

answered Mar 2, 2010 at 19:45

Minor
221 2 7

I think he'll need to temporarily stop writes while running the script. – Ztyx Mar 12, 2014 at 9:08

This still works great. The tools have been renamed though and it's now called pt-table-sync. Actually, though, I've found that their pt-slave-restart tool works like magic. – mlerley Oct 9, 2015 at 19:02

6

Here is what I typically do when a mysql slave gets out of sync. I have looked at mk-table-sync but thought the Risks section was scary looking.

On Master:

```
SHOW MASTER STATUS
```

The outputted columns (File, Position) will be of use to us in a bit.

On Slave:

```
STOP SLAVE
```

Then dump the master db and import it to the slave db.

Then run the following:

```
CHANGE MASTER TO
  MASTER_LOG_FILE='[File]',
  MASTER_LOG_POS=[Position];
START SLAVE;
```

Where [File] and [Position] are the values outputted from the "SHOW MASTER STATUS" ran above.

Hope this helps!

Share Improve this answer Follow

answered Mar 19, 2010 at 17:30

Bryson
1,796 1 11 4

6 This seems broken, since apparently you don't `FLUSH TABLES WITH READ LOCK;` before you `SHOW MASTER STATUS` and dump the master database. I think this might result in e.g. duplicate key errors on the slave since you effectively set the master status to a point in time before the dump was taken, so you'll replay history that's already included in the dump. (If you do things in the order you described.) – KajMagnus Apr 4, 2014 at 10:54

Following up on David's answer...

5

Using `SHOW SLAVE STATUS\G` will give human-readable output.

Share Improve this answer Follow

answered Apr 22, 2013 at 14:07

Greg Ackerson
61 1 2

Any way to page the output? – Josiah Mar 1, 2016 at 16:36

'pager more' I think will do it – Ian Mar 27, 2018 at 7:29

3

Master:

```
mysqldump -u root -p --all-databases --master-data | gzip > /tmp/dump.sql.gz
```

`scp master:/tmp/dump.sql.gz slave:/tmp/` Move dump file to slave server

Slave:

```
STOP SLAVE;
```

```
zcat /tmp/dump.sql.gz | mysql -u root -p
```

```
START SLAVE;
SHOW SLAVE STATUS;
```

NOTE:
On master you can run `SET GLOBAL expire_logs_days = 3` to keep binlogs for 3 days in case of slave issues.

Share Improve this answer Follow

edited Oct 10, 2019 at 13:20

answered Oct 10, 2019 at 13:13

M Rostami
3,925 1 35 38

Here is a complete answer that will hopefully help others...

2

I want to setup mysql replication using master and slave, and since the only thing I knew was that it uses log file(s) to synchronize, if the slave goes offline and gets out of sync, in theory it should only need to connect back to its master and keep reading the log file from where it left off, as user malonso mentioned.

So here are the test result after configuring the master and slave as mentioned by: http://dev.mysql.com/doc/refman/5.0/en/replication-howto.html ...

Provided you use the recommended master/slave configuration and don't write to the slave, he and I where right (as far as mysql-server 5.x is concerned). I didn't even need to use "START SLAVE;", it just caught up to its master. But there is a default 88000 something retries every 60 second so I guess if you exhaust that you might have to start or restart the slave. Anyways, for those like me who wanted to know if having a slave going offline and back up again requires manual intervention.. no, it doesn't.

Maybe the original poster had corruption in the log-file(s)? But most probably not just a server going off-line for a day.

pulled from /usr/share/doc/mysql-server-5.1/README.Debian.gz which probably makes sense to non debian servers as well:

```
* FURTHER NOTES ON REPLICATION
==============================
If the MySQL server is acting as a replication slave, you should not
set --tmpdir to point to a directory on a memory-based filesystem or to
a directory that is cleared when the server host restarts. A replication
slave needs some of its temporary files to survive a machine restart so
that it can replicate temporary tables or LOAD DATA INFILE operations. If
files in the temporary file directory are lost when the server restarts,
replication fails.
```

you can use something sql like: ***show variables like 'tmpdir';*** *to find out.*

Share Improve this answer Follow

edited Dec 3, 2011 at 5:21

Brock Adams
89.7k 22 227 292

answered Oct 5, 2011 at 2:07

bksunday
715 5 21

Will both the database sync automatically between each other? Eg i Write to Master, Will Slave get updated automatically? – TransformBinary Dec 16, 2013 at 13:56

Adding to the popular answer to include this error:

2

```
"ERROR 1200 (HY000): The server is not configured as slave; fix in config file or
with CHANGE MASTER TO",
```

Replication from slave in one shot:

In one terminal window:

```
mysql -h <Master_IP_Address> -uroot -p
```

After connecting,

```
RESET MASTER;
FLUSH TABLES WITH READ LOCK;
SHOW MASTER STATUS;
```

The status appears as below: Note that position number varies!

```
+------------------+----------+--------------+------------------+
| File             | Position | Binlog_Do_DB | Binlog_Ignore_DB |
+------------------+----------+--------------+------------------+
| mysql-bin.000001 |      98  | your_DB      |                  |
+------------------+----------+--------------+------------------+
```

Export the dump similar to how he described "**using another terminal**"!

Exit and connect to your own DB(which is the slave):

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

The type the below commands:

```
STOP SLAVE;
```

Import the Dump as mentioned (in another terminal, of course!) and type the below commands:

```
RESET SLAVE;
CHANGE MASTER TO
  MASTER_HOST = 'Master_IP_Address',
  MASTER_USER = 'your_Master_user', // usually the "root" user
  MASTER_PASSWORD = 'Your_MasterDB_Password',
  MASTER_PORT = 3306,
  MASTER_LOG_FILE = 'mysql-bin.000001',
  MASTER_LOG_POS = 98; // In this case
```

Once logged, set the server_id parameter (usually, for new / non-replicated DBs, this is not set by default),

```
set global server_id=4000;
```

Now, start the slave.

```
START SLAVE;
SHOW SLAVE STATUS\G;
```

The output should be the same as he described.

```
Slave_IO_Running: Yes
Slave_SQL_Running: Yes
```

Note: Once replicated, the master and slave share the same password!

Share Improve this answer Follow

edited Jun 17, 2013 at 6:29

answered May 22, 2013 at 4:44

curlyreggie
1,530 4 21 31

Will both the database sync automatically between each other? Eg i Write to Master, Will Slave get updated automatically? – TransformBinary Dec 16, 2013 at 13:55

1

sometimes you just need to give the slave a kick too

try

```
stop slave;
reset slave;
start slave;
show slave status;
```

quite often, slaves, they just get stuck guys :)

Share Improve this answer Follow

edited Mar 7, 2019 at 19:47

Deepak Mahakale
22.5k 10 66 86

answered Oct 15, 2014 at 16:07

The Masked Coder of Crapness
101 1

...and monitor `Position` on the master using `show master status;` against `Exec_Master_Log_Pos:` on the Slave using `show slave status \G` . The slave should catch up to the master. Worked for me using mysql 5.6 just now after a short network outage. – Mike S Jun 12, 2015 at 15:25

15 This is misleading, once you reset the slave, the salve totally lost knowledge about where the master is. – user1663023 Dec 16, 2015 at 9:47

Rebuilding the slave using LVM

1

Here is the method we use to rebuild MySQL slaves using Linux LVM. This guarantees a consistent snapshot while requiring very minimal downtime on your master.

Set innodb max dirty pages percent to zero on the master MySQL server. This will force MySQL to write all the pages to the disk which will significantly speed up the restart.

```
set global innodb_max_dirty_pages_pct = 0;
```

To monitor the number of dirty pages run the command

```
mysqladmin ext -i10 | grep dirty
```

Once the number stop decreasing you have reach the point to continue. Next reset the master to clear the old bin logs / relay logs:

```
RESET MASTER;
```

Execute lvdisplay to get LV Path

```
lvdisplay
```

Output will look like this

```
--- Logical volume ---
LV Path                /dev/vg_mysql/lv_data
LV Name                lv_data
VG Name                vg_mysql
```

Shutdown the master database with command

```
service mysql stop
```

Next take a snaphot, mysql_snapshot will be the new logical volume name. If binlogs are place on the OS drive those need to be snapshot as well.

```
lvcreate --size 10G --snapshot --name mysql_snapshot /dev/vg_mysql/lv_data
```

Start master again with command

```
service mysql start
```

Restore dirty pages setting to the default

```
set global innodb_max_dirty_pages_pct = 75;
```

Run lvdisplay again to make sure the snapshot is there and visible

```
lvdisplay
```

Output:

```
--- Logical volume ---
LV Path                /dev/vg_mysql/mysql_snapshot
LV Name                mysql_snapshot
VG Name                vg_mysql
```

Mount the snapshot

```
mkdir /mnt/mysql_snapshot
mount /dev/vg_mysql/mysql_snapshot /mnt/mysql_snapshot
```

If you have an existing MySQL slave running you need to stop it

```
service mysql stop
```

Next you need to clear MySQL data folder

```
cd /var/lib/mysql
rm -fr *
```

Back to master. Now rsync the snapshot to the MySQL slave

```
rsync --progress -harz /mnt/mysql_snapshot/ targethostname:/var/lib/mysql/
```

Once rsync has completed you may unmount and remove the snapshot

```
umount /mnt/mysql_snapshot
lvremove -f /dev/vg_mysql/mysql_snapshot
```

Create replication user on the master if the old replication user doesn't exist or password is unknown

```
GRANT REPLICATION SLAVE on *.* to 'replication'@'[SLAVE IP]' identified by
'YourPass';
```

Verify that /var/lib/mysql data files are owned by the mysql user, if so you can omit the following command:

```
chown -R mysql:mysql /var/lib/mysql
```

Next record the binlog position

```
ls -laF | grep mysql-bin
```

You will see something like

```
..
-rw-rw----     1 mysql mysql  1073750329 Aug 28 03:33 mysql-bin.000017
-rw-rw----     1 mysql mysql  1073741932 Aug 28 08:32 mysql-bin.000018
-rw-rw----     1 mysql mysql   963333441 Aug 28 15:37 mysql-bin.000019
-rw-rw----     1 mysql mysql    65657162 Aug 28 16:44 mysql-bin.000020
```

Here the master log file is the highest file number in sequence and bin log position is the file size. Record these values:

```
master_log_file=mysql-bin.000020
master_log_post=65657162
```

Next start the slave MySQL

```
service mysql start
```

Execute change master command on the slave by executing the following:

```
CHANGE MASTER TO
master_host="10.0.0.12",
master_user="replication",
master_password="YourPass",
master_log_file="mysql-bin.000020",
master_log_pos=65657162;
```

Finally start the slave

```
SLAVE START;
```

Check slave status:

```
SHOW SLAVE STATUS;
```

Make sure Slave IO is running and there are no connection errors. Good luck!

I recently wrote this on my blog which is found here... There are few more details there but the story is the same.

http://www.juhavehnia.com/2015/05/rebuilding-mysql-slave-using-linux-lvm.html

Share Improve this answer Follow

edited Oct 30, 2021 at 23:40

BenMorel
33.7k 49 178 314

answered May 21, 2015 at 20:06

Juha Vehnia
29 3

-1

We are using master-master replication technique of MySQL and if one MySQL server say 1 is removed from the network it reconnects itself after the connection are restored and all the records that were committed in the in the server 2 which was in the network are transferred to the server 1 which has lost the connection after restoration. Slave thread in the MySQL retries to connect to its master after every 60 sec by default. This property can be changed as MySQL ha a flag "master_connect_retry=5" where 5 is in sec. This means that we want a retry after every 5 sec.

But you need to make sure that the server which lost the connection show not make any commit in the database as you get duplicate Key error Error code: 1062

Share Improve this answer Follow

answered Sep 15, 2018 at 7:46

parag gupta
parag **84** 1 5