MARIADB REPLICATION BACKUP

Sommaire

[1. Replication 1](#_Toc414615090)

[a. Setup 1](#_Toc414615091)

[b. Bin file management 4](#_Toc414615092)

[2. Backup 7](#_Toc414615093)

[a. Full online backup 8](#_Toc414615094)

[b. Incremental online backup 8](#_Toc414615095)

[c. Point in time backup 8](#_Toc414615096)

[3. Restore 9](#_Toc414615097)

[a. from a full 9](#_Toc414615098)

[b. From an Incr 9](#_Toc414615099)

[c. options. 10](#_Toc414615100)

# Replication

# Setup

<https://mariadb.com/kb/en/mariadb/setting-up-replication/>

La mise en réplication en gros (master : p12, slave : p11) :

/etc/my.cnf.d/server.cnf sur le master (p12), relance du moteur :

[mariadb]

log-basename=frmsazr-p12

log-bin

binlog-format=row

server\_id=12

/etc/my.cnf.d/server.cnf sur le slave (p11), relance du moteur :

[mariadb]

server\_id=11

read-only=true

Côté master, en root, créer le user de réplication :

GRANT REPLICATION SLAVE ON \*.\* TO 'maria\_slave'@'192.168.2.4' IDENTIFIED BY '%D\*WN28dCNFBl(' ;

FLUSH PRIVILEGES ;

Côté master toujours, figer la base, noter le nom et la position du binlog, et prendre un backup :

FLUSH TABLES WITH READ LOCK;

show master status;

+------------------------+----------+--------------+------------------+

| File                   | Position | Binlog\_Do\_DB | Binlog\_Ignore\_DB |

+------------------------+----------+--------------+------------------+

| frmsazr-p12-bin.000001 |      892 |              |                  |

+------------------------+----------+--------------+------------------+

<< faire un backup complet sans arrêter cette session ! >>

unlock tables;

C’est tout pour le master. Transférer le dump vers le slave, et restorer sur le slave :

mysql -u root -p < /tmp/dump.sql

Se connecter au slave, lui dire où est son master et à partir d’où appliquer les logs :

CHANGE MASTER TO

MASTER\_HOST='192.168.1.4',

MASTER\_USER='maria\_slave',

MASTER\_PASSWORD='%D\*WN28dCNFBl(',

MASTER\_PORT=3306,

MASTER\_LOG\_FILE='frmsazr-p12-bin.000001',

MASTER\_LOG\_POS=892,

MASTER\_CONNECT\_RETRY=10;

Et démarrer le tout:

start slave;

Et c’est parti.

A voir: purge des binlogs, switchover/failover, détails de la proc de backup sur le slave.

MariaDB [(none)]> show slave status \G

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Slave\_IO\_State: Waiting for master to send event

Master\_Host: 192.168.1.4

Master\_User: maria\_slave

Master\_Port: 3306

Connect\_Retry: 10

Master\_Log\_File: frmsazr-p12-bin.000001

Read\_Master\_Log\_Pos: 491560

Relay\_Log\_File: frmsazr-p11-relay-bin.000004

Relay\_Log\_Pos: 489644

Relay\_Master\_Log\_File: frmsazr-p12-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: 491560

Relay\_Log\_Space: 489947

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

Master\_SSL\_Verify\_Server\_Cert: No

Last\_IO\_Errno: 0

Last\_IO\_Error:

Last\_SQL\_Errno: 0

Last\_SQL\_Error:

Replicate\_Ignore\_Server\_Ids:

Master\_Server\_Id: 12

Master\_SSL\_Crl:

Master\_SSL\_Crlpath:

Using\_Gtid: No

Gtid\_IO\_Pos:

1 row in set (0.00 sec)

pt-slave-delay -p 'G2IZOM=nU3#>lh' --delay 1m --interval 15s --run-time 10m

# Bin file management

<https://mariadb.com/kb/en/mariadb/using-and-maintaining-the-binary-log/>

In order not to have big bin files, we should use parameters:

expire\_logs\_days = 2

max\_binlog\_size = 100M

In /etc/my.cnf.d.server.cnf

If we need to purge bin file, you should check on slave, that replication is OK (see show slave STATUS \G).

MariaDB [(none)]> SHOW BINARY LOGS ;

+------------------------+------------+

| Log\_name | File\_size |

+------------------------+------------+

| frmsazr-p12-bin.000001 | 367 |

| frmsazr-p12-bin.000002 | 382 |

| frmsazr-p12-bin.000003 | 48885 |

| frmsazr-p12-bin.000004 | 1002038752 |

| frmsazr-p12-bin.000005 | 332 |

+------------------------+------------+

5 rows in set (0.00 sec)

MariaDB [(none)]> PURGE BINARY LOGS BEFORE '2015-03-19 00:00:00' ;

Query OK, 0 rows affected (0.06 sec)

MariaDB [(none)]> SHOW BINARY LOGS ;

+------------------------+------------+

| Log\_name | File\_size |

+------------------------+------------+

| frmsazr-p12-bin.000003 | 48885 |

| frmsazr-p12-bin.000004 | 1002038752 |

| frmsazr-p12-bin.000005 | 332 |

+------------------------+------------+

3 rows in set (0.00 sec)

Example I want to delete frmsazr-p12-bin.000004:

MariaDB [(none)]> SHOW BINARY LOGS ;

+------------------------+------------+

| Log\_name | File\_size |

+------------------------+------------+

| frmsazr-p12-bin.000001 | 367 |

| frmsazr-p12-bin.000002 | 382 |

| frmsazr-p12-bin.000003 | 48885 |

| frmsazr-p12-bin.000004 | 1002038752 |

| frmsazr-p12-bin.000005 | 332 |

+------------------------+------------+

5 rows in set (0.00 sec)

I go on slave p11:

MariaDB [(none)]> show slave status \G

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Slave\_IO\_State: Waiting for master to send event

Master\_Host: 192.168.1.4

Master\_User: maria\_slave

Master\_Port: 3306

Connect\_Retry: 10

Master\_Log\_File: frmsazr-p12-bin.000005

Read\_Master\_Log\_Pos: 332

Relay\_Log\_File: frmsazr-p11-relay-bin.000014

Relay\_Log\_Pos: 541

Relay\_Master\_Log\_File: frmsazr-p12-bin.000005

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: 332

Relay\_Log\_Space: 844

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

Master\_SSL\_Verify\_Server\_Cert: No

Last\_IO\_Errno: 0

Last\_IO\_Error:

Last\_SQL\_Errno: 0

Last\_SQL\_Error:

Replicate\_Ignore\_Server\_Ids:

Master\_Server\_Id: 12

Master\_SSL\_Crl:

Master\_SSL\_Crlpath:

Using\_Gtid: No

Gtid\_IO\_Pos:

1 row in set (0.00 sec)

The master log file is Master\_Log\_File: frmsazr-p12-bin.000005. So I can remove frmsazr-p12-bin.000004 with :

MariaDB [(none)]> PURGE BINARY LOGS TO 'frmsazr-p12-bin.000005' ;

Query OK, 0 rows affected (0.37 sec)

MariaDB [(none)]> SHOW BINARY LOGS ;

+------------------------+-----------+

| Log\_name | File\_size |

+------------------------+-----------+

| frmsazr-p12-bin.000005 | 332 |

+------------------------+-----------+

1 row in set (0.00 sec)

It will keep only bin file from file frmsazr-p12-bin.000005 (so if I have a bin file generated after bin file frmsazr-p12-bin.000005 as frmsazr-p12-bin.000006 for instance, it will keep frmsazr-p12-bin.000005 and frmsazr-p12-bin.000006).

# Backup

For backup we use innobackupexe instead of xtrabackup because it offers more possibilities.

[innobackupex](http://www.percona.com/doc/percona-xtrabackup/2.2/innobackupex/innobackupex_script.html)

a wrapper script that provides functionality to backup a whole *MySQL* database instance with [MyISAM](http://www.percona.com/doc/percona-xtrabackup/2.2/glossary.html" \l "term-myisam), [InnoDB](http://www.percona.com/doc/percona-xtrabackup/2.2/glossary.html" \l "term-innodb), and [XtraDB](http://www.percona.com/doc/percona-xtrabackup/2.2/glossary.html" \l "term-xtradb)tables.

[xtrabackup](http://www.percona.com/doc/percona-xtrabackup/2.2/xtrabackup_bin/xtrabackup_binary.html)

a compiled *C* binary, which copies only [InnoDB](http://www.percona.com/doc/percona-xtrabackup/2.2/glossary.html" \l "term-innodb) and [XtraDB](http://www.percona.com/doc/percona-xtrabackup/2.2/glossary.html" \l "term-xtradb) data

<http://www.percona.com/doc/percona-xtrabackup/2.2/>

We need to create user in database:

Htpp://dev.mysql.com/doc/refman/5.0/en/stop-slave.html

CREATE USER 'bkpuser'@'localhost' IDENTIFIED BY 'eUiyDsnnmRzLZARAx7XT';

GRANT RELOAD, LOCK TABLES, REPLICATION CLIENT, SUPER ON \*.\* TO 'bkpuser'@'localhost';

FLUSH PRIVILEGES;

FLUSH PRIVILEGES ;

# Full online backup

For full backup we do :

backup\_cmd="innobackupex --incremental --user=${BACKUP\_USER} --password=${PASSWORD} ${BACKUP\_DIR\_NAME} --SLAVE-INFO --SAFE-SLAVE-BACKUP --incremental-basedir=${BACKUP\_DIR}/last\_full\_backup --no-timestamp "

prepare\_cmd="innobackupex --apply-log --redo-only ${BACKUP\_DIR}/last\_full\_backup"

# Incremental online backup

For incremental :

backup\_cmd="innobackupex --incremental --user=${BACKUP\_USER} --password=${PASSWORD} ${BACKUP\_DIR\_NAME} --SLAVE-INFO --SAFE-SLAVE-BACKUP --incremental-basedir=${BACKUP\_DIR}/last\_full\_backup --no-timestamp "

prepare\_cmd="innobackupex --apply-log --redo-only ${BACKUP\_DIR}/last\_full\_backup"

We can compress backup but you will have to do preparation before restoring and you can do incremental after…

Example :

innobackupex --stream=tar --user=bkpuser --password=eUiyDsnnmRzLZARAx7XT --SLAVE-INFO --SAFE-SLAVE-BACKUP /var/lib/backups | gzip - > /var/lib/backups/full\_20150319\_161509.tgz

|  |  |  |
| --- | --- | --- |
|  | Size of backup | Duration |
| Non compress full | 2.5GB | 2 min 25 |
| Compress full (tar gunzip) | 761 MB | 8min 11s |

# Bin LOG backup

<http://www.percona.com/blog/2014/03/28/innodb-redo-log-archiving/>

<http://www.percona.com/blog/2012/01/18/backing-up-binary-log-files-with-mysqlbinlog/>

#!/bin/sh

source $1

cd $BACKUPDIR

echo "Backup dir: $BACKUPDIR "

while :

do

LASTFILE=`ls -1 $BACKUPDIR|grep -v orig|tail -n 1`

TIMESTAMP=`date +%s`

FILESIZE=$(stat -c%s "$LASTFILE")

if [ $FILESIZE -gt 0 ]; then

echo "Backing up last binlog"

mv $LASTFILE $LASTFILE.orig$TIMESTAMP

fi

touch $LASTFILE

echo "Starting live binlog backup"

$MBL --raw --read-from-remote-server --stop-never --host $MYSQLHOST --port $MYSQLPORT -u $MYSQLUSER -p$MYSQLPASS $LASTFILE

echo "mysqlbinlog exited with $? trying to reconnect in $RESPAWN seconds."

sleep $RESPAWN

done

config file:

MBL=/opt/mysql5.6/usr/bin/mysqlbinlog

MYSQLHOST=10.10.10.10

MYSQLPORT=3306

MYSQLUSER=replication\_user

MYSQLPASS=replication\_pass

BACKUPDIR=/media/binlogs/server2/

# time to wait before reconnecting after failure

RESPAWN=10

# Restore

# from a full

restore\_full()

{

echo "restore of FULL $1"

log\_date=`date +%Y%m%d\_%H%M%S`

tar\_cmd="tar -czvf ${BACKUP\_DIR}/old\_mysql\_content${log\_date}.tgz ${datadir}"

clean\_cmd="rm -rf ${datadir}/\*"

restore\_cmd="innobackupex --copy-back ${BACKUP\_DIR}/${1}"

chown\_cmd="chown -R mysql:mysql ${datadir}"

launch\_cmd "${tar\_cmd}" || exit $?

launch\_cmd "${clean\_cmd}"|| exit $?

launch\_cmd "${restore\_cmd}"|| exit $?

launch\_cmd "${chown\_cmd}"|| exit $?

}

# From an Incr

restore\_inc()

{

echo "restore of INC $1"

log\_date=`date +%Y%m%d\_%H%M%S`

tar\_cmd="tar -czvf ${BACKUP\_DIR}/old\_mysql\_content${log\_date}.tgz ${datadir}"

clean\_cmd="rm -rf ${datadir}/\*"

chown\_cmd="chown -R mysql:mysql ${datadir}"

from\_lsn=$(grep from\_lsn ${BACKUP\_DIR}/${1}/xtrabackup\_checkpoints | cut -d= -f2 )

full\_backup\_name=$(dirname $(grep "to\_lsn =${from\_lsn}" ${BACKUP\_DIR}/full\*/xtrabackup\_checkpoints | cut -d: -f1))

if [[ -d ${full\_backup\_name} ]]; then

cp\_cmd="cp -rp ${full\_backup\_name} ${BACKUP\_DIR}/restore${log\_date} "

prepare\_cmd2="innobackupex --apply-log --redo-only ${BACKUP\_DIR}/restore${log\_date} --incremental-dir=${BACKUP\_DIR}/${1}"

restore\_cmd="innobackupex --copy-back ${BACKUP\_DIR}/restore${log\_date}"

else

log\_failure\_msg "UNABLE TO FIND FULL BACKUP FOR INC BACKUP ${1}: ${full\_backup\_name} DOES NOT EXIST"

fi

launch\_cmd "${tar\_cmd}"|| exit $?

launch\_cmd "${clean\_cmd}"|| exit $?

launch\_cmd "${cp\_cmd}"|| exit $?

launch\_cmd "${prepare\_cmd2}"|| exit $?

launch\_cmd "${restore\_cmd}"|| exit $?

launch\_cmd "${chown\_cmd}"|| exit $?

rm -rf ${BACKUP\_DIR}/restore${log\_date}

}

# options.

When you restore a slave replicate database, you should setup the replication see 1Set up.

Look at the content of the file xtrabackup\_binlog\_info, it will be something like:

TheSlave$ cat /var/lib/mysql/xtrabackup\_binlog\_info

TheMaster-bin.000001 481

But you should have some error :

MariaDB [(none)]> CHANGE MASTER TO

    -> MASTER\_HOST='192.168.1.4',

    -> MASTER\_USER='maria\_slave',

    -> MASTER\_PASSWORD='%D\*WN28dCNFBl(',

    -> MASTER\_PORT=3306,

    -> MASTER\_LOG\_FILE='frmsazr-p12-bin.000001',

    -> MASTER\_LOG\_POS=892,

    -> MASTER\_CONNECT\_RETRY=10;

Query OK, 0 rows affected (0.14 sec)

MariaDB [(none)]> start slave;

Query OK, 0 rows affected (0.01 sec)

MariaDB [(none)]> show slave status \G

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

               Slave\_IO\_State:

                  Master\_Host: 192.168.1.4

                  Master\_User: maria\_slave

                  Master\_Port: 3306

                Connect\_Retry: 10

              Master\_Log\_File: frmsazr-p12-bin.000001

          Read\_Master\_Log\_Pos: 892

               Relay\_Log\_File: frmsazr-p11-relay-bin.000001

                Relay\_Log\_Pos: 4

        Relay\_Master\_Log\_File: frmsazr-p12-bin.000001

             Slave\_IO\_Running: No

            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: 892

              Relay\_Log\_Space: 248

              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

Master\_SSL\_Verify\_Server\_Cert: No

                Last\_IO\_Errno: 1236

                Last\_IO\_Error: Got fatal error 1236 from master when reading data from binary log: 'Client requested master to start replication from impossible position; the first event 'frmsazr-p12-bin.000001' at 892, the last event read from 'frmsazr-p12-bin.000001' at 4, the last byte read from 'frmsazr-p12-bin.000001' at 4.'

               Last\_SQL\_Errno: 0

               Last\_SQL\_Error:

  Replicate\_Ignore\_Server\_Ids:

             Master\_Server\_Id: 12

               Master\_SSL\_Crl:

           Master\_SSL\_Crlpath:

                   Using\_Gtid: No

                  Gtid\_IO\_Pos:

1 row in set (0.00 sec)

Which you can solve like this :

MariaDB [(none)]> stop slave ;

Query OK, 0 rows affected (0.02 sec)

MariaDB [(none)]> CHANGE MASTER TO MASTER\_LOG\_FILE='frmsazr-p12-bin.000001' , MASTER\_LOG\_POS=4;

Query OK, 0 rows affected (0.43 sec)

MariaDB [(none)]> start slave;

Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> show slave status \G

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

               Slave\_IO\_State: Waiting for master to send event

                  Master\_Host: 192.168.1.4

                  Master\_User: maria\_slave

                  Master\_Port: 3306

                Connect\_Retry: 10

              Master\_Log\_File: frmsazr-p12-bin.000001

          Read\_Master\_Log\_Pos: 318

               Relay\_Log\_File: frmsazr-p11-relay-bin.000002

                Relay\_Log\_Pos: 611

        Relay\_Master\_Log\_File: frmsazr-p12-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: 318

              Relay\_Log\_Space: 914

              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

Master\_SSL\_Verify\_Server\_Cert: No

                Last\_IO\_Errno: 0

                Last\_IO\_Error:

               Last\_SQL\_Errno: 0

               Last\_SQL\_Error:

  Replicate\_Ignore\_Server\_Ids:

             Master\_Server\_Id: 12

               Master\_SSL\_Crl:

           Master\_SSL\_Crlpath:

                   Using\_Gtid: No

                  Gtid\_IO\_Pos:

1 row in set (0.00 sec)