Using MySQL 5.6 Global Transaction IDs in Production

Stéphane Combaudon
August 27th, 2014

Agenda

- Introduction to GTIDs
- Daily DBA tasks
- Typical issues
- High availability solutions with GTIDs

Introduction to GTIDs

What is a GTID? (1)

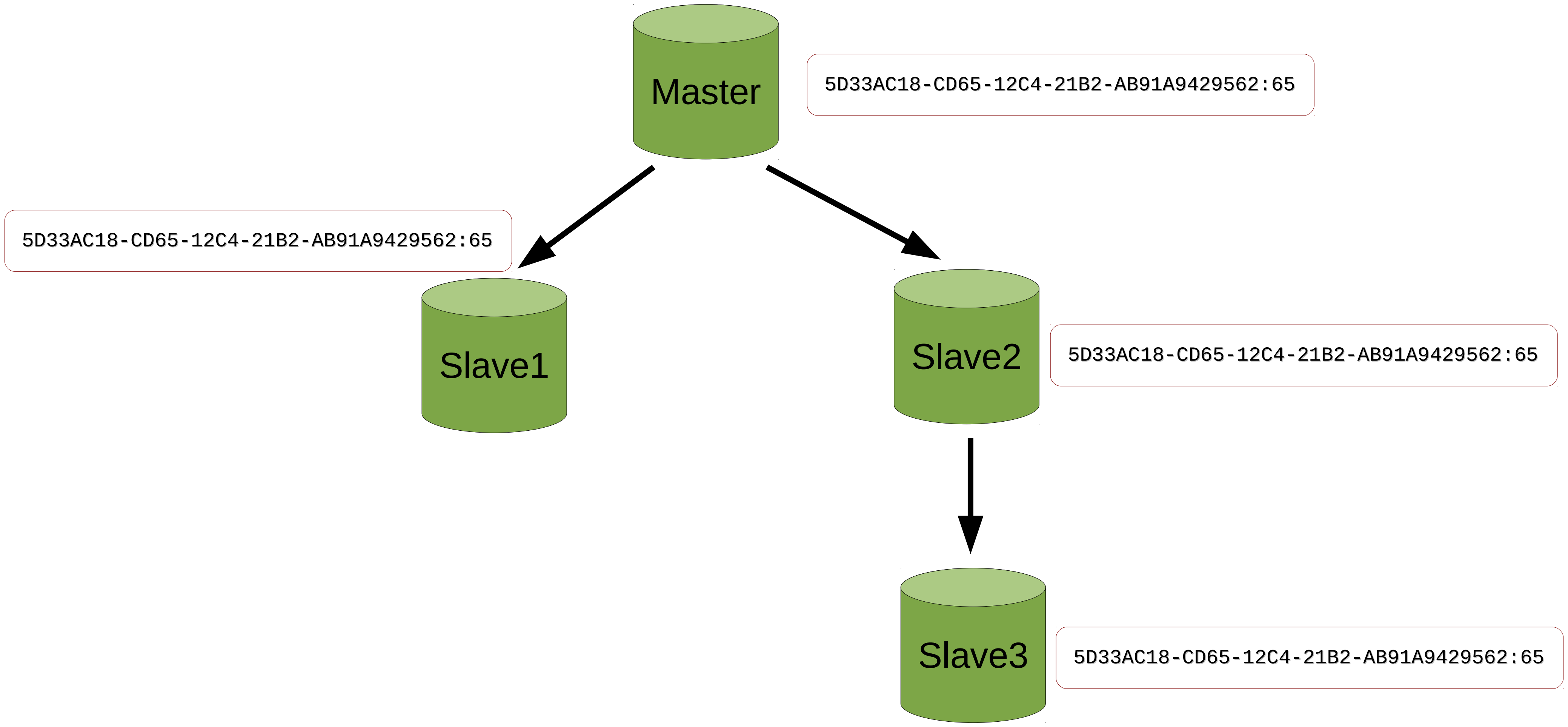
- Unique identifier of a transaction across all servers of a replication setup
- Available from MySQL 5.6
- Main goal: being able to change the replication topology easily
- MariaDB 10 has another implementation that is not compatible
  - Work is in progress to allow replication from MySQL 5.6 to MariaDB 10

What is a GTID? (2)

- 2 parts
  - `source_id:transaction_id`
  - `3E11FA47-71CA-11E1-9E33-C80AA9429562:1`
- Mapping to actual binlog file/position is kept in memory
- A sequence of GTIDs
  - `source_id:trx_start-trx_stop`
  - `3E11FA47-71CA-11E1-9E33-C80AA9429562:1-5`

Finding the position of an event

- Now that's easy, same for all servers!!


Master
5D33AC18-CD65-12C4-21B2-AB91A9429562:65
5D33AC18-CD65-12C4-21B2-AB91A9429562:65
Slave1
Slave2
5D33AC18-CD65-12C4-21B2-AB91A9429562:65
Slave3
5D33AC18-CD65-12C4-21B2-AB91A9429562:65

Transaction ordering

- Transaction counter is per instance, not global
- Say we have `xxx:101` and `xxx:102`, which came first?
  - `xxx:101`
- Say we have `xxx:101` and `yyy:102`, which came first?
  - We don't know

Enabling GTIDs

- Add following settings in my.cnf on all servers
  - `gtid_mode = ON`
  - `log_bin`
  - `log-slave-updates`
  - `enforce-gtid-consistency`
- Then restart all servers at the same time
  - Yes, it's mandatory :(
  - Will be improved in 5.7
  - Booking.com and Facebook have patches for online migration

Using GTID replication

- Once GTIDs are enabled on all servers, run
  - `CHANGE MASTER TO …, MASTER_AUTO_POSITION = 1`

- `MASTER_LOG_POS` and MASTER_LOG_FILE are no longer needed

Replication protocol

- When slave connects to the master
  - Position-based replication
    - Master sends all transactions from the given offset
  - GTID-based replication
    - Sends the range of GTIDs it has executed
    - Master sends back all other transactions
    - Rule: a transaction with a given GTID can only execute once
- More on that new replication protocol later

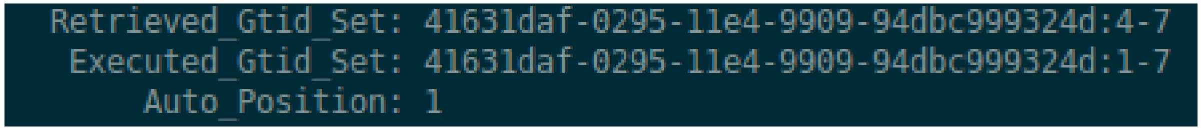
Daily DBA tasks

Provisioning a slave

- mysqldump
  - `--master-data` now includes GTID information
  - Reload the dump and run `CHANGE MASTER TO … MASTER_AUTO_POSITION=1`
- Percona XtraBackup
  - `xtrabackup_binlog_info` contains GTID information
  - After moving the backup, run `SET GLOBAL gtid_purged="XYZ"`
  - Then run `CHANGE MASTER TO … MASTER_AUTO_POSITION=1`

Checking replication status (1)

- New columns for `SHOW SLAVE STATUS`

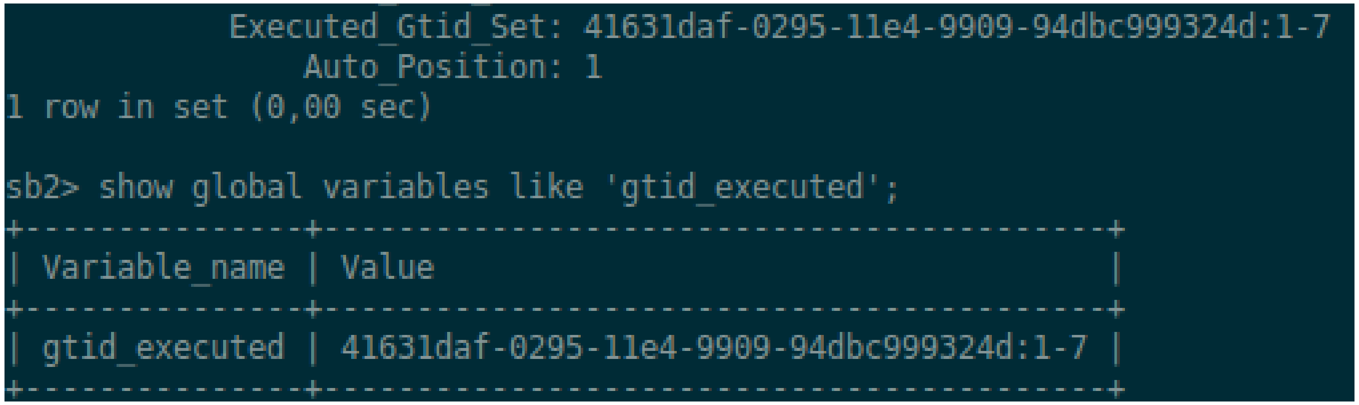


```
  Retrieved_Gtid_Set: 41631daf-0295-11e4-9909-94dbc999324d:4-7
   Executed_Gtid_Set: 41631daf-0295-11e4-9909-94dbc999324d:1-7
       Auto_Position: 1
```

- `Retrieved_Gtid_Set`: GTIDs received by the slave, cleared after a server restart
- `Executed_Gtid_Set`: List of GTIDs executed. Here last executed transaction is `41631daf-0295-11e4-9909-94dbc999324d:7`
- `Auto_position`: 1 if GTID-based replication is in use

Checking replication status (2)

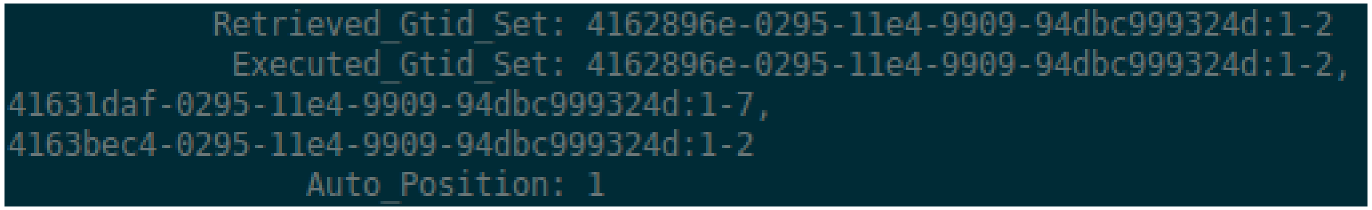
- `gtid_executed`
  - Set of executed GTIDs, identical to `Executed_Gtid_Set`



```
             Executed_Gtid_Set: 41631daf-0295-11e4-9909-94dbc999324d:1-7
                 Auto_Position: 1
1 row in set (0,00 sec)

sb2> show global variables like 'gtid_executed';
+---------------+------------------------------------------+
| Variable_name | Value                                    |
+---------------+------------------------------------------+
| gtid_executed | 41631daf-0295-11e4-9909-94dbc999324d:1-7 |
+---------------+------------------------------------------+
```

- After several failovers, can be more complex



```
           Retrieved_Gtid_Set: 4162896e-0295-11e4-9909-94dbc999324d:1-2
            Executed_Gtid_Set: 4162896e-0295-11e4-9909-94dbc999324d:1-2,
41631daf-0295-11e4-9909-94dbc999324d:1-7,
4163bec4-0295-11e4-9909-94dbc999324d:1-2
                Auto_Position: 1
```

Skipping transactions (1)

- `sql_skip_slave_counter = N` no longer works
  - Because of the new replication protocol, the transaction would automatically come back
  - It throws an error if you try to use it anyway
- But there's a solution!
  - Execute a fake trx with the GTID you want to skip
  - New replication protocol makes sure the real trx will not be executed

Skipping transactions (2)

- How to skip transaction `XXXX:NN`?
  - `STOP SLAVE;`
  - `SET gtid_next = 'XXXX:NN';`
  - `BEGIN;COMMIT;        # Fake transaction!`
  - `SET gtid_next=automatic;`
  - `START SLAVE;`

Typical issues

Errant transactions

- What if you execute a trx locally on a slave?
  - It generates its own GTID
  - If the slave is promoted, trx is sent to all the servers

- That can bite on failover
  - Trx is not desired: sorry, now it is everywhere
  - Trx is no longer in the binlogs: sorry, this triggers a replication error

Detecting errant transactions

- `Executed_Gtid_Set` of any slave should always be a subset of `Executed_Gtid_Set` of master
  - `GTID_SUBSET()` can be used for this check
- If `SELECT GTID_SUBSET(`*`slave_set`*`,`*`master_set`*`)` returns 0, you have errant transactions
- Then use `GTID_SUBTRACT(master_set,slave_set)` to identify them

Fixing errant transactions

- Inject an empty transaction on all other servers of the topology
- If you have to run local transactions, use `SET sql_log_bin = 0`

Holes in the GTID sequence

- Holes are not allowed, but there are bugs
  - http://bugs.mysql.com/bug.php?id=71575
  - http://bugs.mysql.com/bug.php?id=71376 (fixed in 5.6.18)
- That can lead to issues similar to those hit with errant transactions
- No tool is currently checking holes

I/O performance issues

- `log_bin` + `log_slave_updates` adds some I/O overhead

- Mapping between GTID and actual position is kept in memory
  - On initial connect, dump threads has to reverse scan the master's binlogs
  - This can be expensive if the slave is far behind

High Availability solutions with GTIDs

MySQL Utilities

- Set of Python scripts to ease administration of MySQL servers
- Free and open source, developed by Oracle
- http://dev.mysql.com/doc/workbench/en/mysql-utilities.html

Overview of mysqlfailover

- Health monitoring and automatic failover
  - Target topology: 1 master, N slaves

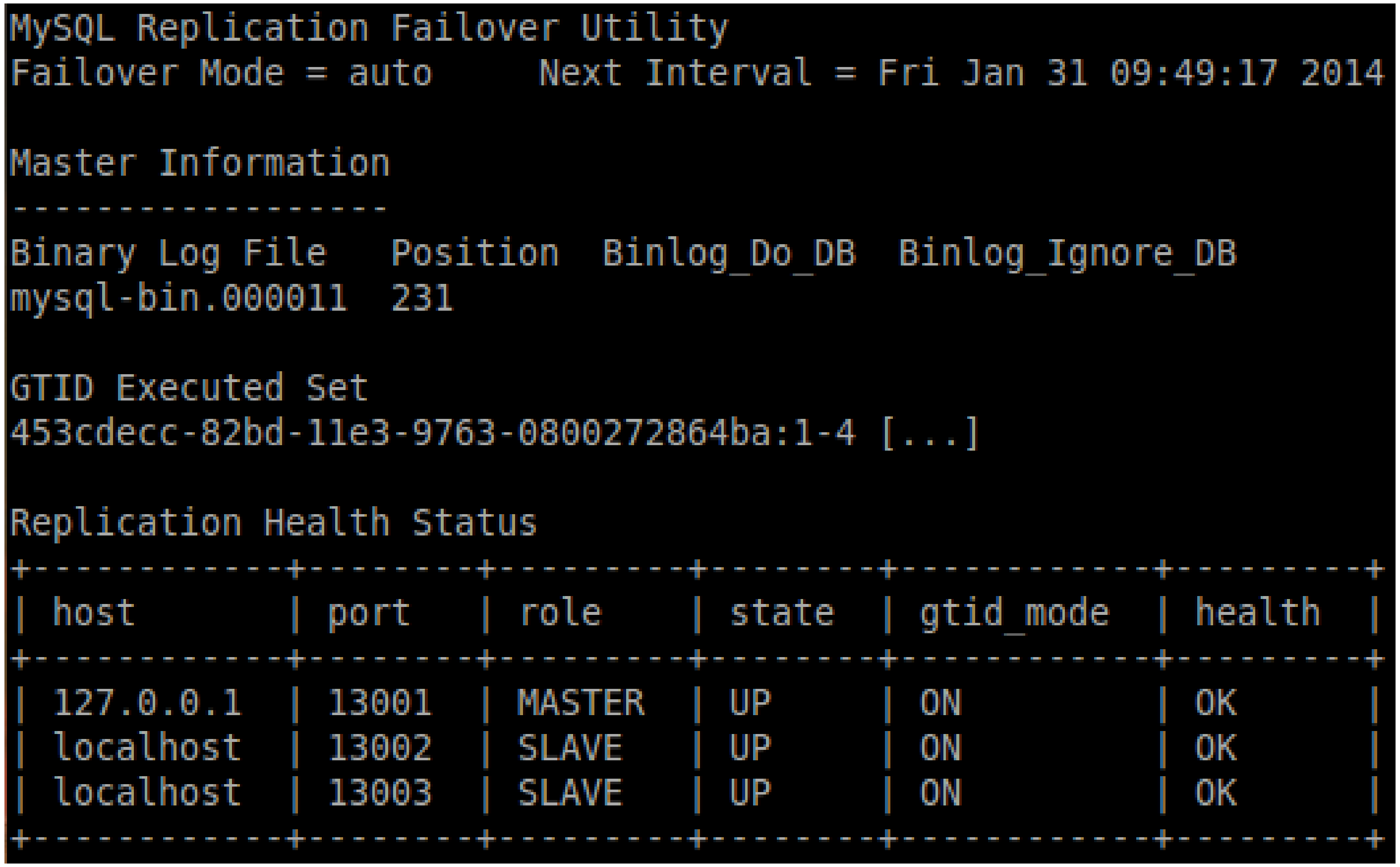
- A few MySQL settings are required
  - `--log-slave-updates, --enforce-gtid-consistency, gtid_mode = ON`
  - `--report-host, --report-port`
  - `--master-info-repository=TABLE`

Existing modes

- Elect
  - Chooses a candidate from a list. If none can be promoted, exits with an error
- Auto (default)
  - Same as elect, but if no candidate is suitable, any other slave can be promoted
- Fail
  - Perform health monitoring, exits with an error if the master fails

Example of execution

```
mysqlfailover --discover-slaves-login=root:root \
--master=root:root@127.0.0.1:13001 auto
```



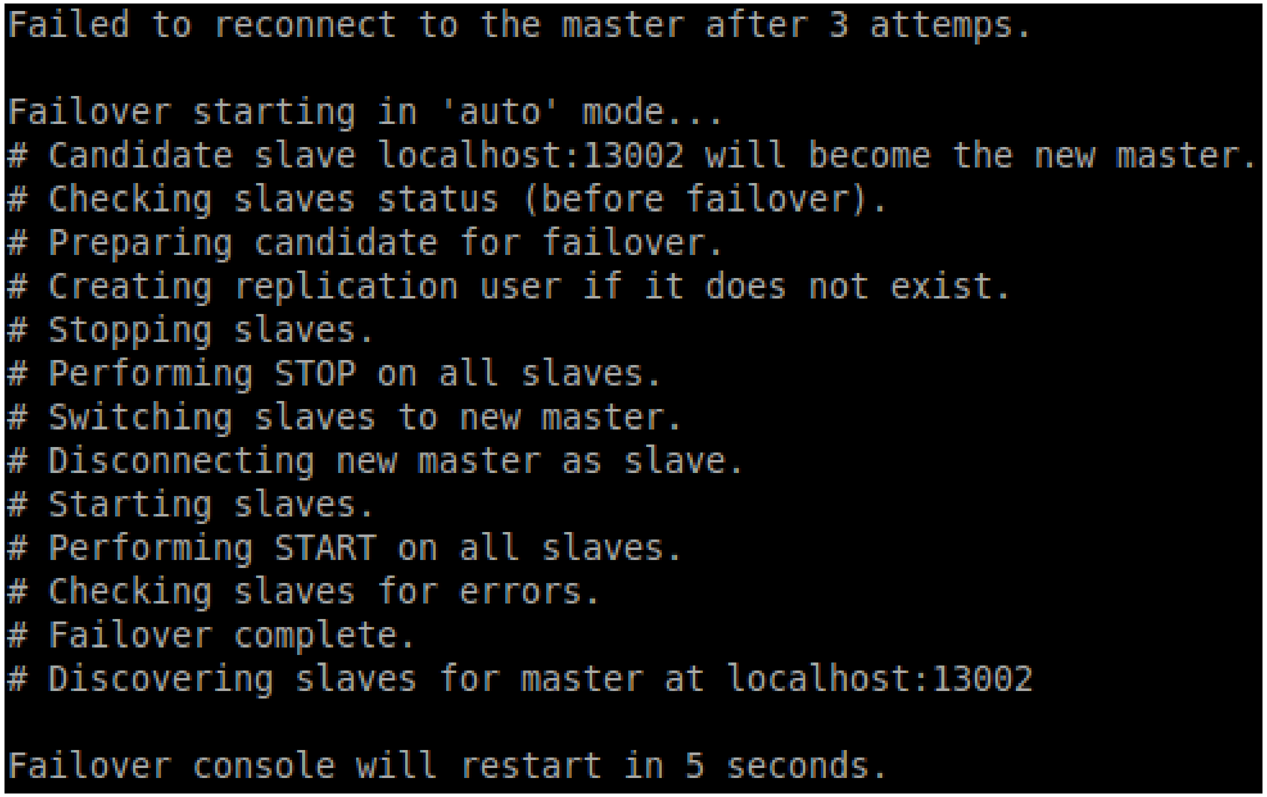
```
MySQL Replication Failover Utility
Failover Mode = auto     Next Interval = Fri Jan 31 09:49:17 2014

Master Information
------------------
Binary Log File   Position  Binlog_Do_DB  Binlog_Ignore_DB
mysql-bin.000011  231

GTID Executed Set
453cdecc-82bd-11e3-9763-0800272864ba:1-4 [...]

Replication Health Status
+------------+--------+---------+--------+------------+---------+
| host       | port   | role    | state  | gtid_mode  | health  |
+------------+--------+---------+--------+------------+---------+
| 127.0.0.1  | 13001  | MASTER  | UP     | ON         | OK      |
| localhost  | 13002  | SLAVE   | UP     | ON         | OK      |
| localhost  | 13003  | SLAVE   | UP     | ON         | OK      |
+------------+--------+---------+--------+------------+---------+
```

If the master fails...

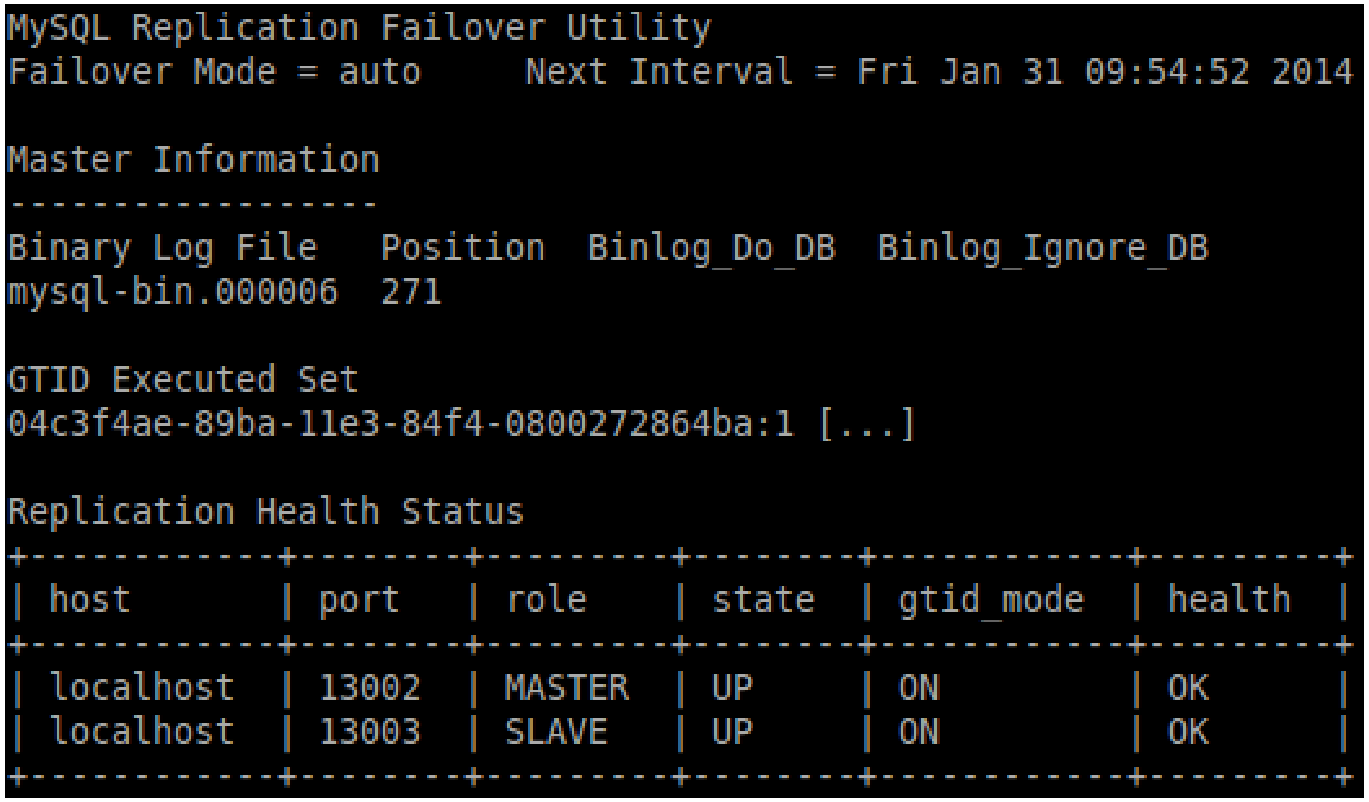


```
Failed to reconnect to the master after 3 attemps.

Failover starting in 'auto' mode...
# Candidate slave localhost:13002 will become the new master.
# Checking slaves status (before failover).
# Preparing candidate for failover.
# Creating replication user if it does not exist.
# Stopping slaves.
# Performing STOP on all slaves.
# Switching slaves to new master.
# Disconnecting new master as slave.
# Starting slaves.
# Performing START on all slaves.
# Checking slaves for errors.
# Failover complete.
# Discovering slaves for master at localhost:13002

Failover console will restart in 5 seconds.
```

When failover is done



```
MySQL Replication Failover Utility
Failover Mode = auto     Next Interval = Fri Jan 31 09:54:52 2014

Master Information
------------------
Binary Log File   Position  Binlog_Do_DB  Binlog_Ignore_DB
mysql-bin.000006  271

GTID Executed Set
04c3f4ae-89ba-11e3-84f4-0800272864ba:1 [...]

Replication Health Status
+------------+---------+----------+---------+-------------+---------+
| host       | port    | role     | state   | gtid_mode   | health  |
+------------+---------+----------+---------+-------------+---------+
| localhost  | 13002   | MASTER   | UP      | ON          | OK      |
| localhost  | 13003   | SLAVE    | UP      | ON          | OK      |
+------------+---------+----------+---------+-------------+---------+
```

Limitations

- Monitoring node is not highly available
  - Closely monitor the monitoring node!
  - Manual failover with `mysqlrpladmin` may be preferred

- Errant transactions can prevent failover
  - Use `--pedantic` to get an error when starting mysqlfailover
  - Fix manually

Manual failover with mysqlrpladmin

- Planned promotion (switchover)

```
mysqlrpladmin --master=root:root@127.0.0.1:13002 \
--new-master=root:root@127.0.0.1:13001 \
--discover-slaves-login=root:root --demote-master \
switchover
```

- Unplanned promotion (failover)

```
mysqlrpladmin
--slaves=root:root@127.0.0.1:13002,root:root@127.0.0.
1:13003 --candidates=root:root@localhost:13002
failover
```

MySQL Fabric

- Available from MySQL Utilities 1.4
- Not limited to HA
- http://www.mysql.fr/products/enterprise/fabric.html
  - “Extensive framework for managing farms of MySQL servers”
- We had a previous webinar on Fabric
  - http://www.percona.com/resources/mysql-webinars/putting-mysql-fabric-use

Other solutions

- Older tools having added support for GTIDs
  - MHA from v0.56
  - Percona Replication Manager (PRM)
- However keep in mind that none of this tools checks errant transactions
  - Using them requires caution

PERCONA LIVE LONDON
MYSQL CONFERENCE 2014

Learn from leading MySQL experts. London, UK. November 3-4, 2014

Early Bird Rates End August 31st

at 11:30pm BST

http://www.percona.com/live/london-2014/

Q&A

Thanks for attending!

stephane.combaudon@percona.com