

MariaDB 10.4 Cluster Rolling Upgrade - Naive Approach

This is a rolling upgrade scenario for upgrading MariaDB 10.3 to MariaDB 10.4 cluster. In the scenario, a 3 node cluster is first started from MariaDB 10.3 source tree by mtr suite galera_3nodes. MariaDB 10.4 nodes will be started from a separate 10.4 source tree, and by using the 10.3 nodes' configuration file so that data directory will be in the 10.3 mtr var directory, but executable and support files will be used from 10.4 source build. Starting 10.4 node will happen in two phases: First we start the node with no wsrep_provider to avoid joining to the running cluster. Mysql_upgrade will be run for this running and isolated node. After that, the node is restarted with proper wsrep_provider pointing to galera 4 library. Upgraded 10.4 node will now join to the cluster.

During the rolling upgrade some mysql SQL load should be targeted to the active nodes in the cluster to see how well replication works inside mixed 10.3-10.4 cluster. Also, new 10.4 features should be open for end user only after all nodes have been upgraded.

Joining of upgraded 10.4 node should happen by IST. If joining has to use SST, the donor will send 10.3 data directory back to 10.4 node, and that practically makes the upgrade void.

Unfortunately, during the first experiment, the IST phase has failed for protocol mismatch. Galera library part will need a fix for this issue.

Prepare MariaDB 10.3 install from MariaDB source directory

```
git checkout 10.3
cmake -DWITH_WSREP=1
make -j4
```

Start 3 MariaDB 10.3 nodes with mtr

```
cd mysql-test
export WSREP_PROVIDER='<path to galera library version 3'
./mtr --suite=galera_3nodes --start-and-exit
Started [mysqld.1 - pid: 23325, winpid: 23325] [mysqld.2 - pid: 23329, winpid: 23329]
[mysqld.3 - pid: 23330, winpid: 23330]
worker[1] Using config for test galera_3nodes.galera_certification_ccc
worker[1] Port and socket path for server(s):
worker[1] mysqld.1 16000 /home/seppo/work/MariaDB/server/mysql-test/var/tmp/mysqld.1.sock
worker[1] mysqld.2 16001 /home/seppo/work/MariaDB/server/mysql-test/var/tmp/mysqld.2.sock
worker[1] mysqld.3 16002 /home/seppo/work/MariaDB/server/mysql-test/var/tmp/mysqld.3.sock
```

At this point, 3 servers should be running

```

seppo@labrador:~/work/MariaDB/server/mysql-test$ ps aux | grep mysqld
seppo  23325  0.6  1.0 2209736 163000 pts/3  S1   14:49   0:05
/home/seppo/work/MariaDB/server/sql/mysqld --defaults-group-suffix=.1
--defaults-file=/home/seppo/work/MariaDB/server/mysql-test/var/my.cnf --log-output=file
--innodb --innodb-cmpmem --innodb-cmp-per-index --innodb-trx --innodb-locks
--innodb-lock-waits --innodb-metrics --innodb-buffer-pool-stats --innodb-buffer-page
--innodb-buffer-page-lru --innodb-sys-columns --innodb-sys-fields --innodb-sys-foreign
--innodb-sys-foreign-cols --innodb-sys-indexes --innodb-sys-tables --innodb-sys-virtual
--core-file --loose-debug-sync-timeout=300
seppo  23329  0.5  1.0 2062000 162964 pts/3  S1   14:49   0:04
/home/seppo/work/MariaDB/server/sql/mysqld --defaults-group-suffix=.2
--defaults-file=/home/seppo/work/MariaDB/server/mysql-test/var/my.cnf --log-output=file
--innodb --innodb-cmpmem --innodb-cmp-per-index --innodb-trx --innodb-locks
--innodb-lock-waits --innodb-metrics --innodb-buffer-pool-stats --innodb-buffer-page
--innodb-buffer-page-lru --innodb-sys-columns --innodb-sys-fields --innodb-sys-foreign
--innodb-sys-foreign-cols --innodb-sys-indexes --innodb-sys-tables --innodb-sys-virtual
--core-file --loose-debug-sync-timeout=300
seppo  23330  0.5  1.0 2061968 162032 pts/3  S1   14:49   0:04
/home/seppo/work/MariaDB/server/sql/mysqld --defaults-group-suffix=.3
--defaults-file=/home/seppo/work/MariaDB/server/mysql-test/var/my.cnf --log-output=file
--innodb --innodb-cmpmem --innodb-cmp-per-index --innodb-trx --innodb-locks
--innodb-lock-waits --innodb-metrics --innodb-buffer-pool-stats --innodb-buffer-page
--innodb-buffer-page-lru --innodb-sys-columns --innodb-sys-fields --innodb-sys-foreign
--innodb-sys-foreign-cols --innodb-sys-indexes --innodb-sys-tables --innodb-sys-virtual
--core-file --loose-debug-sync-timeout=300

```

And using ports:

tcp	0	0	127.0.0.1:631	0.0.0.0:*	LISTEN	-	off (0.00/0/0)
tcp	0	0	127.0.0.1:16000	0.0.0.0:*	LISTEN	23325/mysqld	off (0.00/0/0)
tcp	0	0	127.0.0.1:16001	0.0.0.0:*	LISTEN	23329/mysqld	off (0.00/0/0)
tcp	0	0	127.0.0.1:16002	0.0.0.0:*	LISTEN	23330/mysqld	off (0.00/0/0)
tcp	0	0	0.0.0.0:16003	0.0.0.0:*	LISTEN	23325/mysqld	off (0.00/0/0)
tcp	0	0	0.0.0.0:16006	0.0.0.0:*	LISTEN	23329/mysqld	off (0.00/0/0)
tcp	0	0	0.0.0.0:16009	0.0.0.0:*	LISTEN	23330/mysqld	off (0.00/0/0)

Copy [mysqld.*] groups from var/my.cnf into separate configuration files: mysqld*.cnf, and make following edits

```

wsrep_cluster_address='gcomm://127.0.0.1:16003,127.0.0.1:16006,127.0.0.1:16009'
wsrep_provider=<path to galera 4 library>
basedir=<10.4 source tree>
character-sets-dir=<10.4 source tree>/sql/share/charsets
lc-messages-dir=<10.4 source tree>/sql/share/

```

Start some load for each node

```

MariaDB [test]> create table t (i int primary key auto_increment, j int);
watch "mysql -uroot -h0 -P16000 -e'insert into test.t(j) values(1) ' "
watch "mysql -uroot -h0 -P16001 -e'insert into test.t(j) values(2) ' "

```

```
watch "mysql -uroot -h0 -P16002 -e'insert into test.t(j) values(3)' "
```

Prepare 10.4 version

10.4 node should be started from a separate source tree, which is checkout at the wanted 10.4 branch

```
cd <10.4 source tree>
git checkout bb-10.4-galera4
make -j4
```

10.4 server can be started with sql/mysqld executable and by using the mysqld.*.cnf file from 10.3 mtr tree.

Upgrade node 3

```
kill 23330
../sql/mysqld --defaults-file=<10.3 mtr tree>/var/mysqld.3.cnf --wsrep_provider=none
../client/mysql_upgrade --defaults-file=<10.3 mtr tree>/var/mysqld.3.cnf -uroot -h0 -P16002
kill $(pidof ../sql/mysqld)
export PATH=$PATH:<10.4 source tree>/scripts
```

Join node 3 back to cluster

```
../sql/mysqld --defaults-file=../var/mysqld.3.cnf
```

Look at mysql status to verify that node has joined back and wsrep_cluster_size = 3.

In the test experiment, however, the joining node failed immediately during the IST phase. Problem appears to be in donor side, following error is reported:

```
2019-01-20 20:26:22 0 [ERROR] WSREP: async IST sender failed to serve tcp://127.0.0.1:16007:
invalid protocol version 255, expected 9: 71 (Protocol error)
    at galera/src/ist_proto.hpp:unserialize():151
```

This suggests that the joining node uses wrong protocol version for IST phase. In 10.3 donor side, it appears that joiner would have version -1

Upgrade nodes 1 and 2

Do same upgrade for node 2 and then for node 1. When all nodes have been upgraded, and joined back to cluster, it will be possible to use new Galera 4 features.