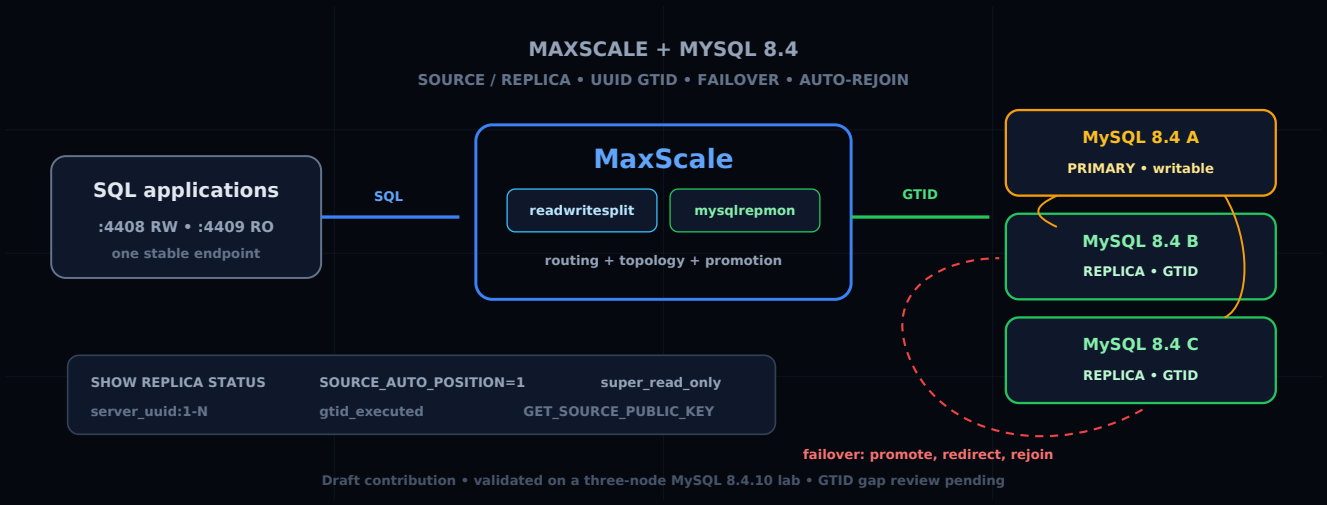
MaxScale وMySQL 8.4: بناء تحويل فشل GTID متوافق مع Source/Replica

Aurélien LEQUOY · 27 يوليو 2026

MAXSCALE | MYSQL | MYSQL-8.4 | REPLICATION | GTID | FAILOVER | HIGH-AVAILABILITY | PROXY



> [illegible] — 27 [illegible] 2026. [illegible] MaxScale [illegible] `mysqlrepmon` [illegible] [illegible] MaxScale. [illegible]

المشكلة في جملة واحدة

راقب MaxScale تاريخياً بنيات نسخ MariaDB / MySQL بواسطة `mariadbmon`. لكن MySQL 8.4 حذف أوامر `SLAVE` / `MASTER` القديمة، ويستخدم GTID مبنية على UUID، ويتطلب صيغة مختلفة لإعادة تهيئة النسخ.

قد تكون النتيجة مضللة: يستمر الوكيل في قبول اتصالات SQL، بينما لم يعد المراقب قادراً على قراءة البنية بصورة صحيحة أو ترقية replica أو إعادة ربط الـprimary السابق.

يجب التمييز بين ثلاث وظائف:

1. **مراقبة** البنية وحالة خيوط النسخ؛
2. **تعديل** البنية أثناء switchover أو failover أو rejoin؛
3. **توجيه** جلسات التطبيقات إلى الـprimary الحالي والـreplicas المتاحة.

يجب أن يكون التوافق مع MySQL 8.4 صحيحاً في المستويات الثلاثة. استبدال `SHOW SLAVE STATUS` بـ `SHOW REPLICA STATUS` وحده لا يكفي.

لماذا يكسر MySQL 8.4 الافتراضات التاريخية

قدّم MySQL 8.0.22 مصطلحات Source/Replica مع إبقاء عدد من الأسماء البديلة القديمة. يُكمل MySQL 8.4 هذا الانتقال: حُذفت الأوامر المهملة.

العملية	MariaDB / الصيغة القديمة	MySQL 8.4
قراءة حالة replica	`SHOW SLAVE STATUS`	`SHOW REPLICA STATUS`
تهيئة source	`CHANGE MASTER TO`	`CHANGE REPLICATION SOURCE TO`
بدء النسخ	`START SLAVE`	`START REPLICA`
إيقاف النسخ	`STOP SLAVE`	`STOP REPLICA`
مسح التهيئة	`RESET SLAVE`	`RESET REPLICA`
قراءة حالة binlog	`SHOW MASTER STATUS`	`SHOW BINARY LOG STATUS`
تصفير GTID	`RESET MASTER`	`RESET BINARY LOGS AND GTIDS`

تتغير أيضاً أسماء الأعمدة المعادة:

الاسم القديم	الاسم في MySQL 8.4
`Master_Host`	`Source_Host`
`Master_Port`	`Source_Port`
`Master_Server_Id`	`Source_Server_Id`
`Slave_IO_Running`	`Replica_IO_Running`
`Slave_SQL_Running`	`Replica_SQL_Running`
`Seconds_Behind_Master`	`Seconds_Behind_Source`
`Master_Log_File`	`Source_Log_File`
`Read_Master_Log_Pos`	`Read_Source_Log_Pos`
`Exec_Master_Log_Pos`	`Exec_Source_Log_Pos`

الاسم القديم	الاسم في MySQL 8.4
`Channel_Name`	`Channel_Name`

يفشل المراقب الذي ما زال يرسل `SHOW SLAVE STATUS` فوراً بخطأ صياغة. أما المراقب الذي يرسل الاستعلام الصحيح لكنه لا يزال يبحث عن `Slave_IO_Running`، فسيستنتج خطأً أن النسخ لا يعمل.

الاختلاف الحقيقي: نموذج GTID

الفرق الأكثر جوهرية ليس مفردات SQL، بل طريقة تمثيل المعاملات.

GTID في MariaDB

تمثل MariaDB قيمة GTID بالشكل:

```
domain_id-server_id-sequence
```

مثال:

```
0-101-7842
```

النطاق جزء من النموذج. يستطيع مراقب MariaDB مقارنة هذه المواضع ويستخدم، من بين آليات أخرى، `MASTER_USE_GTID`.

GTID في MySQL

يمثل MySQL مجموعة GTID بواسطة UUID الخادم والفواصل:

```
155fa720-7623-11f1-9a78-bc241174cab8:1-76
```

قد يحتوي التاريخ على عدة UUID وفواصل غير متصلة:

```
155fa720-7623-11f1-9a78-bc241174cab8:1-10:20-30,
7fd8c42a-7630-11f1-99af-bc241174cab8:1-8
```

يعرض MySQL، من بين قيم أخرى:

- `@@global.server_uuid` لتعريف المصدر؛
- `@@global.gtid_executed` للمعاملات المطبقة؛

- `@@global.gtid_purged` لقيم GTID التي حُذفت binlogs الخاصة بها؛
- `Retrieved_Gtid_Set` داخل `SHOW REPLICA STATUS` للمعاملات المستلمة.

يستخدم تغيير المصدر التموضع التلقائي:

```
CHANGE REPLICATION SOURCE TO
    SOURCE_HOST = '10.0.0.12',
    SOURCE_PORT = 3306,
    SOURCE_USER = 'repl',
    SOURCE_PASSWORD = 'secret',
    SOURCE_AUTO_POSITION = 1,
    GET_SOURCE_PUBLIC_KEY = 1
FOR CHANNEL '';
```

يختار الخادم بعد ذلك نقطة الاستئناف الصحيحة من مجموعات GTID من دون تحديد ملف binlog وموضع داخله.

لماذا نحتاج إلى وحدة `mysqlrepmon`

إلى جانب `mariadbmon`، تضيف المساهمة مراقباً مخصصاً لـMySQL. تعيد استخدام محرك اتخاذ القرار وإدارة العنقود المجرّب، لكنها تستبدل الأجزاء الخاصة بالخادم:

- قراءة `SHOW REPLICA STATUS` أو عمدة `Source_*` / `Replica_*`؛
- قراءة `server_uuid` و `gtid_executed` و `gtid_purged`؛
- استخدام `CHANGE REPLICATION SOURCE TO ... SOURCE_AUTO_POSITION=1`؛
- أوامر `START` و `STOP` و `RESET REPLICA ... FOR CHANNEL`؛
- تفعيل `super_read_only=1` عند خفض دور خادم؛
- استخدام `RESET BINARY LOGS AND GTIDS` لأمر reset اليدوي؛
- فحص `enforce_gtid_consistency` و `log_replica_updates`.

تحتفظ الوحدة بالمعلمات التشغيلية المعروفة في `mariadbmon`: `auto_failover` و `auto_rejoin` و `enforce_read_only_slaves`، إضافة إلى أوامر switchover وfailover وrejoin.

البنية المرجعية

تضم البنية المختبرة ثلاثة خوادم MySQL 8.4 وعقدة MaxScale واحدة:

```
                              +---------------------+
تطبيقات SQL ----------------> | MaxScale            |
المنفذ 4408 (RW)              | readwritesplit      |
المنفذ 4409 (RO)              | mysqlrepmon         |
                              +----------+----------+
                                         |
                        topology, health, GTID, routing
                                         |
               +-------------------------+-------------------------+
               |                         |                         |
       +-------v--------+        +-------v--------+        +-------v--------+
       | mysql84-a      |        | mysql84-b      |        | mysql84-c      |
       | Primary        |------->| Replica        |        | Replica        |
       | 10.0.0.11      |------->| 10.0.0.12      |        | 10.0.0.13      |
       +----------------+        +----------------+        +----------------+
```

لا تعرف التطبيقات عنوان الـprimary مطلقاً، بل تستخدم listener في MaxScale. يقرر المراقب أي خادم يحمل دور `Master` الداخلي في MaxScale؛ ويرسل `readwritesplit` عمليات الكتابة إليه ويوزع القراءات.

> [illegible] *MaxScale* [illegible]. [illegible] *failover* [illegible] *SQL* [illegible]

1. إعداد كل خادم MySQL 8.4

يجب أن يستطيع كل خادم مرشح للترقية إنتاج binlogs الخاصة به وتسجيل المعاملات المستلمة من النسخ.

التهيئة الأساسية:

```
[mysqld]
server_id=101
log_bin=mysql-bin
binlog_format=ROW

gtid_mode=ON
enforce_gtid_consistency=ON
log_replica_updates=ON

relay_log_recovery=ON
```

استخدم `server_id` مختلفاً على كل عقدة، مثل `101` و `102` و `103`.

يجب أن يكون `server_uuid` فريداً أيضاً. يُحفظ في ملف `auto.cnf` داخل datadir. إذا نسخت آلة أو datadir، فلا تشغّل عدة خوادم بالقيمة نفسها لـ `server_uuid`.

على الـreplicas:

```
read_only=ON
super_read_only=ON
```

على الـprimary:

```
read_only=OFF
super_read_only=OFF
```

بعد إعادة التشغيل، افحص الثوابت:

```
SELECT
    @@hostname,
    @@server_id,
    @@server_uuid,
    @@gtid_mode,
    @@enforce_gtid_consistency,
    @@log_bin,
    @@log_replica_updates,
    @@read_only,
    @@super_read_only\G

SELECT @@global.gtid_executed\G
SHOW BINARY LOG STATUS\G
SHOW REPLICA STATUS\G
```

يجب أن تملك replica القابلة للترقية `log_bin=1` و `log_replica_updates=1`، وأن يعمل خيط النسخ، وأن يكون التأخر تحت السيطرة.

2. إنشاء حسابات منفصلة

افصل على الأقل بين:

- حساب مراقبة البنية وتعديلها؛

- الحساب الذي تستخدمه الـreplicas لقراءة binlogs؛
- حسابات التطبيقات التي تمر عبر الوكيل.

حساب المراقب

للمراقبة فقط:

```
CREATE USER 'maxscale_mon'@'10.0.0.10'
    IDENTIFIED BY 'a-long-random-password';

GRANT REPLICATION CLIENT
    ON *.* TO 'maxscale_mon'@'10.0.0.10';
```

لتنفيذ failover وswitchover وrejoin، يجب أن يستطيع المراقب أيضاً إيقاف النسخ وإعادة تهيئته، وتغيير متغيرات read-only، والتحكم في الاتصالات:

```
GRANT REPLICATION SLAVE,
      REPLICATION CLIENT,
      PROCESS,
      SUPER,
      SYSTEM_VARIABLES_ADMIN,
      REPLICATION_SLAVE_ADMIN,
      CONNECTION_ADMIN
    ON *.* TO 'maxscale_mon'@'10.0.0.10';
```

ما زالت بعض مسارات التوافق تطلب الصلاحية التاريخية `SUPER`. في النسخة النهائية من الوحدة، أعد تقييم القائمة الدقيقة واحذف أي صلاحية لم تعد ضرورية.

حساب النسخ

```
CREATE USER 'repl'@'10.0.0.%'
    IDENTIFIED BY 'another-long-random-password'
    REQUIRE SSL;

GRANT REPLICATION SLAVE
    ON *.* TO 'repl'@'10.0.0.%';
```

فضّل TLS. عند استخدام مصادقة `caching_sha2_password` الافتراضية في MySQL 8.4 عبر اتصال غير مشفر، يجب أن يحدد أمر تغيير المصدر `GET_SOURCE_PUBLIC_KEY=1` أو مسار المفتاح العام RSA. تضيف الوحدة هذا الخيار عندما لا يكون TLS مفعّلاً.

حساب الخدمة لمصادقة العملاء

الممثل `user` في خدمة MaxScale الحساب الذي تُنفذ به كل استعلامات التطبيقات. بل يتيح لـUser Account Manager في MaxScale تحميل الحسابات والصلاحيات من خوادم backend:

```
CREATE USER 'maxscale_route'@'10.0.0.10'
    IDENTIFIED BY 'route-password';

GRANT SELECT ON mysql.user
    TO 'maxscale_route'@'10.0.0.10';
GRANT SELECT ON mysql.db
    TO 'maxscale_route'@'10.0.0.10';
GRANT SELECT ON mysql.tables_priv
    TO 'maxscale_route'@'10.0.0.10';
GRANT SELECT ON mysql.columns_priv
    TO 'maxscale_route'@'10.0.0.10';
GRANT SELECT ON mysql.procs_priv
    TO 'maxscale_route'@'10.0.0.10';
GRANT SELECT ON mysql.proxies_priv
    TO 'maxscale_route'@'10.0.0.10';
GRANT SHOW DATABASES ON *.*
    TO 'maxscale_route'@'10.0.0.10';
```

يجب أن تبقى حسابات التطبيقات موجودة على خوادم MySQL بالسر نفسه وبصلاحيات متناسقة. وبما أن backend يرى الاتصال قادماً من MaxScale، يجب أن يسمح جزء `@host` بعنوان الوكيل.

3. بناء المساهمة عند SHA قابل لإعادة الإنتاج

لأن طلب السحب ما زال draft، لا تحتوي حزمة MaxScale القياسية على `mysqlrepmon`. للمختبر فقط، ابنِ قيمة SHA العامة الدقيقة:

```
git clone https://github.com/mariadb-corporation/MaxScale.git
cd MaxScale

git fetch https://github.com/Esysteme/MaxScale.git \
  aurelien/mysql-8.4-replication-support
git checkout --detach f61776a5b19cf295636c94a8a3dea6d81fcd61fb

cmake -S . -B build \
```

```
  -DCMAKE_BUILD_TYPE=Release \
  -DCMAKE_INSTALL_PREFIX=/usr/local/maxscale-mysql84

cmake --build build --target mariadbmon mysqlrepmon test_cycle_find -j2
ctest --test-dir build --output-on-failure \
  -R '^test_mariadbmon_cycle_find$'

cmake --build build -j2
sudo cmake --install build
```

يجب أن ينتج البناء الموجّه `libmysqlrepmon.so`. يعيد إصلاح CMake في المساهمة استخدام `LIBSSH_LIBRARY` و `LIBSSH_INCLUDE_DIR` اللذين اكتشفهما MaxScale، بدل افتراض وجود target داخلي باسم `libssh`.

تشغيل البناء المعزول

يمنع البادئ `/usr/local/maxscale-mysql84` استبدال الحزمة المثبتة. في المقابل، تظل خدمة الحزمة `maxscale.service` تشغّل `/usr/bin/maxscale` ولن تحمّل الوحدة الجديدة. في المختبر، أنشئ `/etc/systemd/system/maxscale-mysql84.service`:

```
[Unit]
Description=MaxScale MySQL 8.4 compiled build
After=network-online.target
Wants=network-online.target

[Service]
Type=simple
User=maxscale
Group=maxscale
RuntimeDirectory=maxscale
RuntimeDirectoryMode=0755
ExecStart=/usr/local/maxscale-mysql84/bin/maxscale --nodaemon --config=/etc/maxscale.cnf --
logdir=/var/log/maxscale --datadir=/var/lib/maxscale --cachedir=/var/cache/maxscale --
libdir=/usr/local/maxscale-mysql84/lib/maxscale --piddir=/run/maxscale
Restart=on-failure
RestartSec=5s
LimitNOFILE=65535

[Install]
WantedBy=multi-user.target
```

يجب أن يكون حساب النظام `maxscale` ومجلدات البيانات موجودة؛ وتنشئها عادةً حزمة MaxScale المثبتة لتوفير بيئة التشغيل. بعد ذلك أعد تحميل إعدادات systemd:

```
sudo install -d -o maxscale -g maxscale -m 0755 \
  /var/lib/maxscale /var/cache/maxscale /var/log/maxscale
sudo systemctl daemon-reload
```

4. تهيئة MaxScale

مثال أدنى كامل:

```
[maxscale]
threads=auto
admin_host=127.0.0.1
admin_port=8989

[mysql84-a]
type=server
address=10.0.0.11
port=3306
protocol=MariaDBBackend

[mysql84-b]
type=server
address=10.0.0.12
port=3306
protocol=MariaDBBackend

[mysql84-c]
type=server
address=10.0.0.13
port=3306
protocol=MariaDBBackend

[MySQL84-Monitor]
type=monitor
module=mysqlrepmon
servers=mysql84-a,mysql84-b,mysql84-c
user=maxscale_mon
password=a-long-random-password
```

```
monitor_interval=2000ms

assume_unique_hostnames=true
auto_failover=safe
auto_rejoin=true
enforce_read_only_slaves=true

replication_user=repl
replication_password=another-long-random-password
replication_master_ssl=true

[MySQL84-RW-Service]
type=service
router=readwritesplit
servers=mysql84-a,mysql84-b,mysql84-c
user=maxscale_route
password=route-password

master_reconnection=true
master_failure_mode=fail_on_write

[MySQL84-RW-Listener]
type=listener
service=MySQL84-RW-Service
protocol=MariaDBClient
port=4408

[MySQL84-RO-Service]
type=service
router=readconnroute
servers=mysql84-a,mysql84-b,mysql84-c
user=maxscale_route
password=route-password
router_options=slave

[MySQL84-RO-Listener]
type=listener
service=MySQL84-RO-Service
protocol=MariaDBClient
port=4409
```

تُركت كلمات المرور مكشوفة هنا لسهولة قراءة المثال فقط. في التشغيل الفعلي، استخدم تشفير الأسرار الذي يوفره MaxScale، وصلاحيات صارمة لملف التهيئة، وعملية تدوير موثقة.

يلزم ضبط `assume_unique_hostnames=true` لتمكين `auto_failover` و`auto_rejoin`. لذلك يجب أن يعرض كل backend عنواناً أو hostname فريداً وثابتاً يطابق الـsource المسجل لدى replicas؛ استخدم `private_address` إذا كانت شبكة النسخ تستخدم عنواناً مختلفاً.

لماذا نبدأ بـ`auto_failover=safe`؟ يرفض هذا الوضع العملية إذا اكتشف المراقب أن الترقية ستؤدي بوضوح إلى فقد معاملات. لا يحول النسخ غير المتزامن إلى متزامن، لكنه يضيف حاجز حماية مفيد أثناء التقييم.

لماذا `master_reconnection=true` و`master_failure_mode=fail_on_write`؟ يمكن لجلسة `readwritesplit` الاحتفاظ بسياقها والاتصال بالـprimary الجديد ما دام لم يصل طلب كتابة أثناء نافذة غياب primary ولا توجد معاملة مفتوحة. من دون هذه المعلمات قد ينجح failover قاعدة البيانات، بينما تُقطع كل جلسات التطبيقات.

5. التحقق قبل الاختبار الأول

تحقق من التهيئة ثم شغّل MaxScale:

```
sudo /usr/local/maxscale-mysql84/bin/maxscale \
  --config=/etc/maxscale.cnf \
  --libdir=/usr/local/maxscale-mysql84/lib/maxscale \
  --config-check

sudo systemctl disable --now maxscale.service 2>/dev/null || true
sudo systemctl enable --now maxscale-mysql84.service
systemctl --no-pager --full status maxscale-mysql84.service
```

تأكد من ظهور الوحدة والبنية:

```
maxctrl list modules | grep -E 'mariadbmon|mysqlrepmon'
maxctrl list servers
maxctrl list monitors
maxctrl show monitor MySQL84-Monitor
```

الحالة المتوقعة:

- خادم واحد `Master, Running`؛
- خادمان `Slave, Running`؛

- الخادم بحالة `Down`؛
- monitor فعّال؛
- listener على `4408` و `4409` مفتوحان.

تحقق أيضاً من MySQL مباشرة:

```
mysql -h 10.0.0.12 -e "SHOW REPLICA STATUS\\G"
mysql -h 10.0.0.13 -e "SHOW REPLICA STATUS\\G"
```

الحقول الأساسية:

```
Replica_IO_Running: Yes
Replica_SQL_Running: Yes
Seconds_Behind_Source: 0
Auto_Position: 1
```

6. فهم تسلسل failover

عند اختفاء الـprimary، لا يغيّر المراقب مجرد تسمية:

1. ينتظر عدد الدورات المحدد في `failcount` ويتحقق قدر الإمكان من أن العطل حقيقي؛
2. يقارن حالة الـreplicas ويختار مرشحاً قابلاً للترقية؛
3. يوقف النسخ على المرشح؛
4. يعطل `read_only` على الـprimary الجديد؛
5. يعيد توجيه بقية الـreplicas باستخدام `CHANGE REPLICATION SOURCE TO ... SOURCE_AUTO_POSITION=1`؛
6. يعيد تشغيل خيوط النسخ لديها؛
7. يكتشف `readwritesplit` الدور الجديد ويوجه الكتابات اللاحقة إلى الـprimary الجديد.

لخفض دور primary، يستخدم `mysqlrepmon`:

```
SET GLOBAL super_read_only = 1;
```

هذا أقوى من `read_only=1`: حتى الحساب ذو الصلاحيات الإدارية يجب ألا يستمر عرضاً في الكتابة إلى الـprimary السابق.

7. اختبار switchover مضبوط

قبل محاكاة عطل قاسٍ، تحقق من switchover:

```
maxctrl call command mysqlrepmon switchover \
  MySQL84-Monitor mysql84-b mysql84-a
```

ثم افحص:

```
maxctrl list servers

mysql -h 10.0.0.11 -e \
  "SELECT @@hostname, @@read_only, @@super_read_only, @@global.gtid_executed\\G"
mysql -h 10.0.0.12 -e \
  "SELECT @@hostname, @@read_only, @@super_read_only, @@global.gtid_executed\\G"
mysql -h 10.0.0.13 -e \
  "SELECT @@hostname, @@read_only, @@super_read_only, @@global.gtid_executed\\G"
```

يجب أن يقبل الـprimary الجديد الكتابة. ويجب أن تحمل العقدتان الأخريان `super_read_only=1` وتنسخا منه.

8. اختبار عطل حقيقي من دون تحايل

أنشئ أولاً بيانات عبر MaxScale:

```
CREATE DATABASE maxscale_ha_test;
CREATE TABLE maxscale_ha_test.events (
    id BIGINT UNSIGNED NOT NULL AUTO_INCREMENT,
    source_hostname VARCHAR(255) NOT NULL,
    created_at TIMESTAMP(6) NOT NULL DEFAULT CURRENT_TIMESTAMP(6),
    PRIMARY KEY (id)
) ENGINE=InnoDB;

INSERT INTO maxscale_ha_test.events(source_hostname)
VALUES (@@hostname);
```

تحقق من وجودها على الـreplicas، ثم أوقف الـprimary على مستوى الخدمة أو الآلة. لا تضع الخادم فقط في وضع maintenance داخل MaxScale: فهذا يختبر قراراً إدارياً لا اكتشاف العطل.

أثناء الاختبار:

```
watch -n 1 'maxctrl list servers'
```

وفي طرفية أخرى:

```
journalctl -u maxscale -f
```

معايير النجاح:

- انتخاب primary جديد واحد فقط؛
- توجيه الـreplicas الأخرى إليه؛
- نجاح كتابة عبر listener `4408` بعد الترقية؛
- بقاء القراءات عبر `4409` متسقة؛
- إعادة انضمام الـprimary السابق بوصفه replica بعد عودته؛
- تقارب مجموعات `gtid_executed`.

ما أثبته مختبر MySQL 8.4.10

نُفذ السيناريو على ثلاث عقد MySQL 8.4.10:

- اكتشاف أولي لـprimary وreplicas اثنتين؛
- إيقاف الـprimary؛
- ترقية إحدى الـreplicas؛
- إعادة توجيه الـreplica المتبقية إلى source الجديد؛
- عودة الـprimary السابق وrejoin تلقائي بوصفه replica؛
- failover ثانٍ في الاتجاه المعاكس؛
- تقارب نهائي إلى مجموعة `gtid_executed` نفسها على الخوادم الثلاثة؛
- توجيه صحيح للقراءة والكتابة عبر MaxScale؛
- رفض الكتابات على listener الخاص بالقراءة فقط.

يثبت هذا الاختبار المسار الوظيفي لتواريخ GTID المتصلة. لكنه لا يثبت بعد كل الحالات الحدية المطلوبة للدمج upstream.

المصادقة: لا تخلط بين عطلين

رفض build MaxScale المستخدم في المختبر حسابات التطبيقات التي تستخدم hash `caching_sha2_password` في MySQL 8.4:

```
Stored password hash length is 70 when 40 was expected
```

تأتي الرسالة من authenticator بروتوكول MaxScale المستخدم في ذلك build، الا من مراقب النسخ. قد يكون النسخ والتوجيه سليمين مع فشل مصادقة العميل.

في المختبر، سمح حساب probe مخصص صراحةً يستخدم `mysql_native_password` بعزل المشكلة. ليست هذه توصية عامة: يعطل MySQL 8.4 هذا plugin التاريخي افتراضياً. في الإنتاج استخدم build وauthenticator لـMaxScale متوافقين مع `caching_sha2_password`، أو طريقة مصادقة مدعومة رسمياً، بدل إضعاف تهيئة MySQL عالمياً.

تنطبق القاعدة نفسها على:

```
401 Unauthorized
```

على منفذ REST: تعني أن `maxctrl` لا يملك بيانات الاعتماد الإدارية الصحيحة، وليست دليلاً على تعطل monitor أو توجيه SQL.

جدول التشخيص

العرض	السبب المرجح	التحقق
خطأ صياغة في `SHOW SLAVE STATUS`	وحدة قديمة أو صياغة MariaDB مستخدمة مع MySQL 8.4	سجل MaxScale والاستعلام المرسل فعلياً
خادم من دون دور replica	أعمدة `Replica_*` غير مربوطة	تنفيذ مباشر لـ`SHOW REPLICA STATUS\G`
تعذر ترقية replica	تعطيل `log_bin` أو `log_replica_updates` أو GTID	المتغيرات العامة وسجل monitor
رفض rejoin	معاملات errant أو تاريخ غير متوافق	مقارنة `gtid_executed` / `gtid_purged`
`Stored password hash length is 70...`	authenticator غير متوافق مع `caching_sha2_password`	سجل MaxScale وplugin الحساب
`401 Unauthorized` مع `maxctrl`	بيانات REST خاطئة	تهيئة إدارة MaxScale

العرض	السبب المرجح	التحقق
انقطاع الجلسة رغم الترقية	تعذر إعادة اتصال router أو معاملة مفتوحة أو نفاد تاريخ أوامر الجلسة	`master_reconnection` و `master_failure_mode` وسجل router
primary اثنان قابلان للكتابة	fencing/read-only غير كافٍ أو monitors متنافسة	`super_read_only` والأقفال التعاونية وحالة الشبكة

القيد المعروف: الفجوات في مجموعات GTID

لإعادة استخدام محرك `mariadbmon` الداخلي، تربط المساهمة كل UUID في MySQL بنطاق اصطناعي وتحول فواصله إلى التمثيل الداخلي الحالي.

في الحالة الحالية، تحتفظ بأعلى رقم معاملة لكل UUID. لذلك:

```
uuid:1-10:20-30
```

يُلخص كما لو أن الموضع وصل إلى `30`. والتمثيل بدقة معلومة غياب المعاملات من 11 إلى 19.

كانت التواريخ في المختبر متصلة. في بنية تحوي GTID محقونة أو مفلترة أو محذوفة أو errant، قد يشوه هذا التقريب مقارنة المرشحين.

هذا هو السبب الرئيسي لبقاء طلب السحب draft. قبل اعتبار الوحدة جاهزة للإنتاج، يجب:

- تمثيل الفواصل غير المتصلة أو مقارنتها بصورة صحيحة؛
- إضافة اختبارات unit بعدة UUID وفجوات؛
- تغطية GTID errant والتواريخ المحذوفة؛
- تشغيل CI upstream الكامل؛
- الحصول على مراجعة maintainer في MaxScale.

ما لا يضمنه failover

لا يلغي المراقب خصائص النسخ غير المتزامن.

- **لا ضمان لـRPO صفري**: قد لا تكون معاملة أكدها الـprimary السابق قد وصلت إلى replica.
- **المعاملات المفتوحة**: لا يمكن دائماً نقل جلسة داخل معاملة بلا خطأ.

- **Split-brain**: إذا ظل الـprimary السابق متاحاً لبعض التطبيقات، يلزم fencing شبكي أو نظامي حقيقي.
- **البنى المعقدة**: تحتاج multi-source والنسخ الدائري وrelay إلى استراتيجية خاصة.
- **البيانات المتباعدة**: يجب ألا يستبدل `auto_rejoin` معاملات errant بصمت.
- **وكيل منفرد**: يجب جعل MaxScale متكرراً بصورة مستقلة.

لذلك يقيس اختبار التوافر العالي الجيد أربعة أمور منفصلة: الاكتشاف، والترقية، وتقارب البيانات، واستمرارية التطبيق.

Checklist قبل الإنتاج

- [ ] نسخة احتياطية مجربة وقابلة للاستعادة؛
- [ ] ثلاث قيم فريدة لـ `server_id` و `server_uuid`؛
- [ ] تفعيل GTID و `log_replica_updates` في كل مكان؛
- [ ] TLS بين MaxScale وخوادم backend؛
- [ ] فصل حسابات monitor والنسخ والتطبيقات؛
- [ ] التحقق من `super_read_only=1` على كل replica؛
- [ ] اختبار switchover مضبوط قبل failover قاسٍ؛
- [ ] اختبار كتابة عبر MaxScale بعد الترقية؛
- [ ] اختبار rejoin للـprimary السابق؛
- [ ] مقارنة نهائية لمجموعات `gtid_executed`؛
- [ ] توثيق سلوك معاملات التطبيقات؛
- [ ] اختبار fencing وتكرار MaxScale؛
- [ ] تنبيه عند كل ترقية أو اختلاف GTID؛
- [ ] إنهاء الوحدة upstream ومراجعتها ودعمها قبل الإنتاج.

لماذا ننشر هذه المساهمة

MySQL 8.4 إصدار LTS. لن تعود الأوامر التاريخية. إبقاء الأسماء البديلة إلى الأبد في أدوات المراقبة لا يحل اختلاف نموذج GTID ولا صياغة إعادة التهيئة.

تجعل الوحدة المخصصة الحد واضحاً:

- يبقى `mariadbmon` أصلياً لصياغة MariaDB وGTID الخاصة بها؛

- يحمل `mysqlrepmon` قواعد MySQL 8.x؛
- يظل محرك failover المشترك معاد الاستخدام؛
- تستطيع الاختبارات التحقق من كل عائلة من دون تكثير الشروط الموزعة.

تحتوي PR MaxScale #421 على الشيفرة والتوثيق وأوامر التحقق ونتائج المختبر وقيد GTID المعروف. الغرض من draft هو تحديداً الحصول على مراجعة لهذا التمثيل قبل الادعاء بأن كل تاريخ MySQL آمن.

قراءة إضافية

- MySQL 8.4 — الميزات الجديدة وأوامر النسخ المحذوفة
- MySQL 8.4 — تهيئة النسخ
- MySQL 8.4 — تغيير source و `GET_SOURCE_PUBLIC_KEY`
- MaxScale — معلمات MariaDB Monitor
- MaxScale — router readwritesplit
- تثبيت MySQL 8.4 على Debian 13
- فهم الانتقال من Master/Slave إلى Source/Replica

هل تريد رؤية هذا الدعم داخل MaxScale؟

إذا كان توافق MySQL 8.4 مفيداً لك، فاقرأ المقترح واختبر الفرع وقدّم دعمك مباشرة في طلب السحب. ستساعد الملاحظات من بيئات حقيقية وحالات GTID المعقدة والمراجعات التقنية على تحويل هذا draft إلى مساهمة قابلة للدمج فعلاً:

ادعم PR MaxScale #421