

Restore replica after failure

Categories: Known Issue · Topics: Databases, MariaDB, MySQL, Replication · Context: Mission Pin Code · Exported: 2026-06-22 09:32

Summary

This document assumes that there was a failure with the replica and one of the nodes remained as the healthy node and the other one as unhealthy. Services were stopped in the unhealthy node to avoid further data inconsistencies after the replica failure

Procedure

Introduction

The procedure to restore the replica will consist on the following steps:

- Stop the Radius service in both nodes, to avoid any writing in the database
- Make a data dump on the healthy node and transfer it to the unhealthy node
- Start MySQL service on the unhealthy node before doing the data restore
- Restore the data dump in the unhealthy node
- Restore the replica setup between the 2 nodes
- Start again the Radius service in both backend nodes

Radius **must remain stopped on both nodes** during the entire operation.

Steps

1. Stop Radius on both nodes

Run on both **dfs-telrad-02** and **dfs-telrad-52**:

```
systemctl stop radiusd  
systemctl status radiusd
```

2. Immediately back up the unhealthy node (safety copy)

On the **unhealthy** node:

```
mysqldump -u root -pR4d1uS_P4ss --all-databases --single-transaction --q
| gzip > /root/unhealthy-prewipe-$(date +%F-%H%M).sql.gz
```

3. Validate DB list before wiping anything x

Expected DBs from the backup script:

```
PINCodeApp
minurso
minusca
unrcca
unsmil
unsoa
osasgcp
osesgy
unama
unami
unifil
unidad
umik
unowas
unisfa
binuh
unukr
```

Check for surprises:

```
mysql -u root -pR4d1uS_P4ss -e "SHOW DATABASES;"
```

If you see:

- extra DBs: stop and investigate
- missing DBs: stop and investigate
- system DBs (mysql, information_schema, performance_schema, sys): safe to ignore

4. Generate fresh dumps on the healthy node x

On the **healthy** node:

```
/system-mysql/backup/radius-backup.sh
```

Transfer to the unhealthy node:

```
mkdir -p /tmp/backup-restore
rsync -avz /system-mysql/backup/ crodrigo@dfs-telrad-52:/tmp/backup-restore
```

5. Start MySQL on the unhealthy node

```
systemctl start mariadb
systemctl status mariadb
```

Make sure MySQL is running before the restore

6. Drop and recreate mission DBs on the unhealthy node

Run once:

```
mysql -u root -pR4d1uS_P4ss -e "
DROP DATABASE IF EXISTS PINCodeApp; CREATE DATABASE PINCodeApp;
DROP DATABASE IF EXISTS minurso; CREATE DATABASE minurso;
DROP DATABASE IF EXISTS minusca; CREATE DATABASE minusca;
DROP DATABASE IF EXISTS unrcca; CREATE DATABASE unrcca;
DROP DATABASE IF EXISTS unsmil; CREATE DATABASE unsmil;
DROP DATABASE IF EXISTS unsoa; CREATE DATABASE unsoa;
DROP DATABASE IF EXISTS osasgcp; CREATE DATABASE osasgcp;
DROP DATABASE IF EXISTS osesgy; CREATE DATABASE osesgy;
DROP DATABASE IF EXISTS unama; CREATE DATABASE unama;
DROP DATABASE IF EXISTS unami; CREATE DATABASE unami;
DROP DATABASE IF EXISTS unifil; CREATE DATABASE unifil;
DROP DATABASE IF EXISTS unitad; CREATE DATABASE unitad;
DROP DATABASE IF EXISTS unmik; CREATE DATABASE unmik;
DROP DATABASE IF EXISTS unowas; CREATE DATABASE unowas;
DROP DATABASE IF EXISTS unisfa; CREATE DATABASE unisfa;
DROP DATABASE IF EXISTS binuh; CREATE DATABASE binuh;
DROP DATABASE IF EXISTS unukr; CREATE DATABASE unukr;
"
```

7. Restore all dumps on the unhealthy node

Unzip:

```
cd /tmp/backup-restore gunzip *.gz
```

Restore:

```
for f in *.sql; do
    db=$(basename "$f" .sql)
    echo "Restoring $db ..."
    mysql -u root -pR4d1uS_P4ss "$db" < "$f"
done
``
```

8. Rebuild replication

8.1 On healthy node (get log position)

```
mysql -u root -p -e "SHOW MASTER STATUS\G"
```

8.2 On unhealthy node

```
mysql -u root -pR4d1uS_P4ss
```

```
STOP SLAVE;  
RESET SLAVE ALL;
```

```
CHANGE MASTER TO  
  MASTER_HOST='dfs-telrad-02',  
  MASTER_USER='replica_user',  
  MASTER_PASSWORD='replica_pass',  
  MASTER_LOG_FILE='mysql-bin.000123',  
  MASTER_LOG_POS=456789;
```

```
START SLAVE;
```

8.3 On healthy node (restore master-master)

```
mysql -u root -pR4d1uS_P4ss
```

```
STOP SLAVE;  
RESET SLAVE ALL;
```

```
CHANGE MASTER TO  
  MASTER_HOST='dfs-telrad-52',  
  MASTER_USER='replica_user',  
  MASTER_PASSWORD='replica_pass',  
  MASTER_LOG_FILE='mysql-bin.000456',  
  MASTER_LOG_POS=123456;
```

```
START SLAVE;
```

8.4 Verification on both nodes

```
mysql -u root -p -e "SHOW SLAVE STATUS\G"
```

Check:

- Seconds_Behind_Master = 0
 - Slave_IO_Running = Yes
 - Slave_SQL_Running = Yes
 - Last_Error = empty
-

9. Start Radius on both nodes

```
systemctl start radiusd  
systemctl status radiusd
```