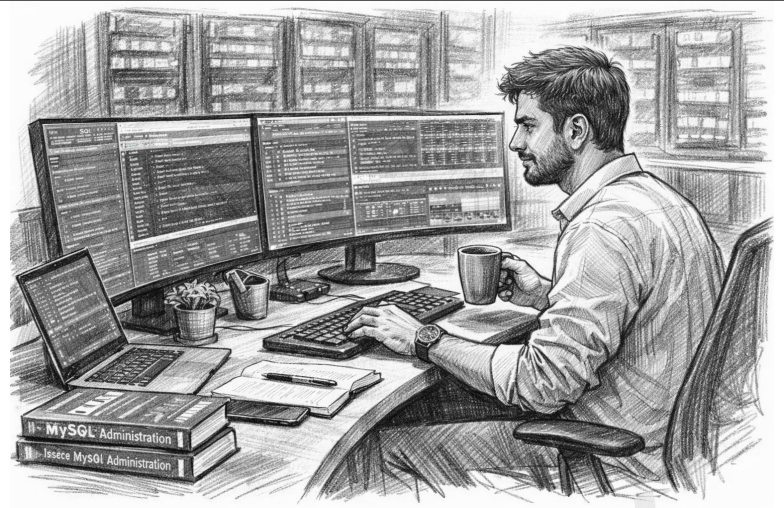


MySQL Day 14

Day 14: Replication & High Availability Basics (Production Level)

Today's topic ensures:

Zero/Low Downtime

Disaster Protection

Read Scalability

Without replication → High business risk ⚠

With replication → Reliable system ✓

Author : Ajit Yadav (MSSQL , MYSQL , POSTGRESQL DBA)

Day 14 Objectives

By the end of today, you will:

✓ Understand MySQL Replication Architecture
✓ Configure Master–Slave Replication
✓ Use GTID Replication
✓ Monitor Replication Health
✓ Handle Replication Errors
✓ Build Basic High Availability

MySQL Master-Slave Replication

What is Replication ?

Replication = Copying data from Primary → Replica.

Primary (Master) → Replica (Slave)
Writes Reads

🔷 Why Companies Use Replication ?

✓ High Availability
✓ Load Balancing
✓ Backup Support
✓ Disaster Recovery
✓ Reporting Server

How Replication Works (Internals)
Replication Flow

Client → Master → Binlog → Slave IO → Relay Log → SQL Thread

Step-by-step:
1 Master writes to Binlog
2 Slave reads Binlog
3 Saves to Relay Log
4 Executes on Slave

If any step breaks → Replication stops ❌

Replication Types

🔹 **Classic (Position-Based)**
Uses:
Binlog File
Position
❌ Error-prone
❌ Manual recovery

🔹 **GTID Replication (Recommended ✓)**
GTID = Global Transaction ID
Each transaction has unique ID.

Benefits:
✓ Auto recovery
✓ Easy failover
✓ Less mistakes
Note : Always use GTID in production 💯

Create a replication user and grant permission to that user .
create user 'repl_user'@'%' identifed by 'abc@1234'

Grant permission for replication to that user :
Grant replication slave on *.* to 'repl_user'@'%';

```
GRANT REPLICATION SLAVE ON *.* TO `repl_user`@`%`
```

Setup GTID Replication (Step-by-Step)
We'll do **Primary + Replica**.

◈ Step 1: Configure Primary (Master)

Edit my.cnf:

```
[mysqld]
server-id=1
log_bin=mysql-bin
gtid_mode=ON
enforce_gtid_consistency=ON
log_slave_updates=ON
```

Restart MySQL.

```
root@db-01:~#
root@db-01:~# sudo nano /etc/mysql/my.cnf
```

```
!includedir /etc/mysql/conf.d/
!includedir /etc/mysql/mysql.conf.d/
[mysqld]
server_id=1
log_bin = mysql-bin
binlog_format = ROW
gtid_mode=ON
enforce_gtid_consistency=ON
log_slave_updates=ON
```

After this changes press
ctrl+o and press enter and press ctrl+x
to exit

```
root@db-01:~# sudo systemctl restart mysql.service
root@db-01:~#
```

◈ Step 2: Configure Replica (Slave)

Edit my.cnf:

```
[mysqld]
server-id=2
log_bin=mysql-bin
gtid_mode=ON
enforce_gtid_consistency=ON
```

Restart MySQL.

```
root@db-02:~#
root@db-02:~# nano /etc/mysql/my.cnf
```

```
!includedir /etc/mysql/conf.d/
!includedir /etc/mysql/mysql.conf.d/
[mysqld]
server_id = 2
log_bin= mysql-bin
binlog_format=ROW
gtid_mode=ON
enforce_gtid_consistency=ON
log_slave_updates=ON
```

After this changes press
ctrl+o and press enter and press ctrl+x
to exit

```
root@db-02:~# systemctl restart mysql.service
root@db-02:~#
```

◈ Step 3: Take Backup from Master

mysqldump -u root -p --all-databases \
--single-transaction --master-data=2 > full.sql

```
root@db-01:/mysql_backup# mysqldump -u root -p --all-databases --single-transaction --master-data=2 > full_database.sql
WARNING: --master-data is deprecated and will be removed in a future version. Use --source-data instead.
Enter password:
```

use below command to take a backup

```
root@db-01:/mysql_backup# mysqldump -u root -p --all-databases --single-transaction --source-data=2 > full_database.sql
Enter password:
Warning: A partial dump from a server that has GTIDs will by default include the GTIDs of all transactions, even those that
 --set-gtid-purged=OFF. To make a complete dump, pass --all-databases --triggers --routines --events.
root@db-01:/mysql_backup#
```

in this command we still facing error but backup has been done the error say's we have enabling the GTID mode so need to take a backup with gtid to restore on slave server so that we have make a errorless replication
below is the updated command .

```
root@db-01:/mysql_backup#
root@db-01:/mysql_backup# mysqldump -u root -p --all-databases --triggers --routines --events --single-transaction --set-gtid-purged=ON > alldatabases_020326.sql
Enter password:
root@db-01:/mysql_backup#
```

```
root@db-01:/mysql_backup# ll
total 64980
drwxr-xr-x  2 root root     4096 Mar  2 13:07 ./
drwxr-xr-x 23 root root     4096 Feb 27 14:14 ../
-rw-r--r--  1 root root  1332513 Mar  2 13:07 alldatabases_020326.sql
```

Backup file genrated now copy that .sql file and paste on slave server and restor there

Copy to Slave.

The file we can copy using file zila or differenet method u can search on google i copied using scp :

```
root@db-01:/mysql_backup# scp alldatabases_020326.sql root@172.18.163.65:/mysql_backup/
root@172.18.163.65's password:
alldatabases_020326.sql                                                    100% 1301KB  11.1MB/s   00:00
root@db-01:/mysql_backup#
```

on secondary server we can see the file are copied

```
root@db-02:/mysql_backup# ll
total 17008
drwxr-xr-x  2 root root    4096 Mar  2 13:17 ./
drwxr-xr-x 21 root root    4096 Dec 29 16:24 ../
-rw-r--r--  1 root root 1332513 Mar  2 13:17 alldatabases_020326.sql
```

◈ Step 4: Restore on Slave

mysql -u root -p < full.sql

```
root@db-02:/mysql_backup# mysql -u root -p < alldatabases_020326.sql
Enter password:
root@db-02:/mysql_backup#
```

◈ Step 5: Configure Replication

On Slave:

CHANGE MASTER TO
MASTER_HOST='master_ip',
MASTER_USER='repl',
MASTER_PASSWORD='pass',
MASTER_AUTO_POSITION=1;

```
mysql> change master to
    -> master_host='172.18.163.64',
    -> master_user='repl_user',
    -> master_password='R3pl@1234',
    -> master_auto_position=1;
Query OK, 0 rows affected, 7 warnings (0.73 sec)

mysql>
```

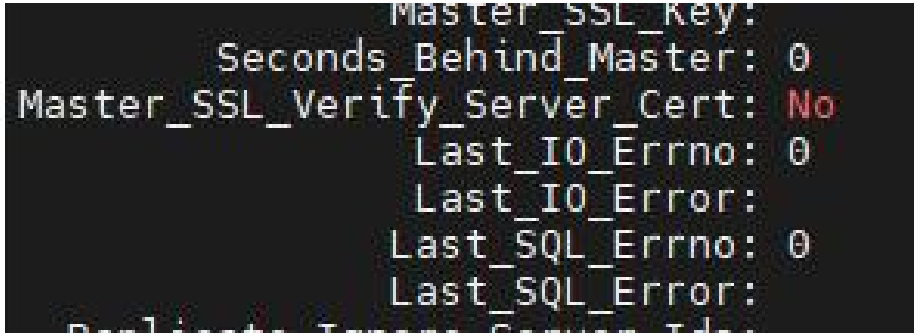
START SLAVE;

```
mysql>
mysql> start slave;
Query OK, 0 rows affected, 1 warning (0.32 sec)

mysql>
```

◈ Step 6: Check Status
SHOW SLAVE STATUS\G

```
mysql> show slave status\G;
*************************** 1. row ***************************
               Slave_IO_State: Waiting for source to send event
                  Master_Host: 172.18.163.64
                  Master_User: repl_user
                  Master_Port: 3306
                Connect_Retry: 60
              Master_Log_File: mysql-bin.000044
          Read_Master_Log_Pos: 392
               Relay_Log_File: db-02-relay-bin.000002
                Relay_Log_Pos: 420
        Relay_Master_Log_File: mysql-bin.000044
             Slave_IO_Running: Yes
            Slave_SQL_Running: Yes
              Replicate_Do_DB:
          Replicate_Ignore_DB:
           Replicate_Do_Table:
       Replicate_Ignore_Table:
```

Must show:

Slave_IO_Running: Yes
Slave_SQL_Running: Yes
If both YES → Working ✓

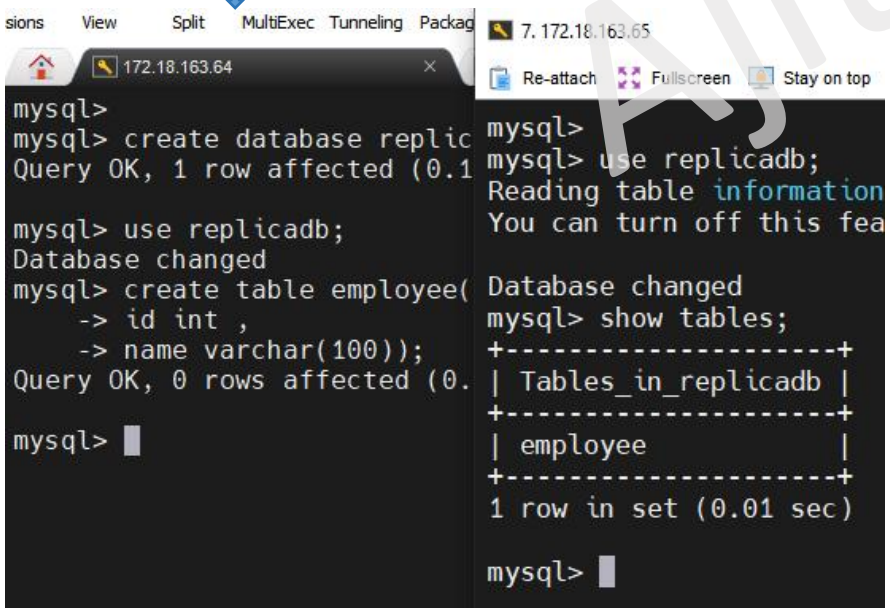
Value	Meaning
0	Perfect ✓
< 10	OK
> 100	Problem ⚠

Monitor Replication (Daily DBA Task)
Check lag:
SHOW SLAVE STATUS\G

Look for :
Seconds_Behind_Master

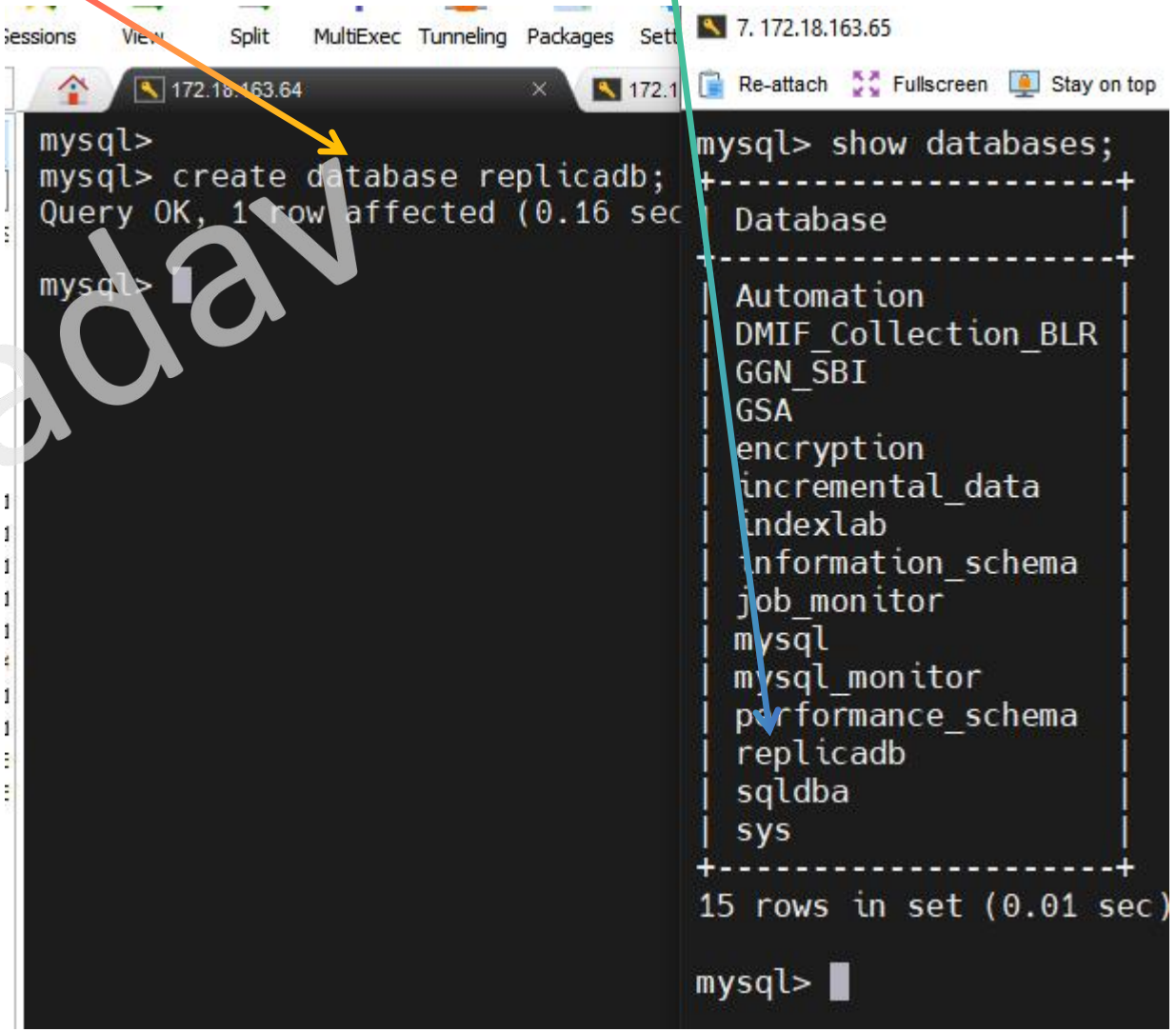
```
                  Master_SSL_Key:
        Seconds_Behind_Master: 0
Master_SSL_Verify_Server_Cert: No
                Last_IO_Errno: 0
                Last_IO_Error:
               Last_SQL_Errno: 0
               Last_SQL_Error:
```

Table replicate as well .



create db on primary

replicate on secondary or slave server



Now server are running as primary and secondary or master slave

But in this we need the secondary serve read_only mode . first we check the server is running read_only or read_write

```
mysql> show variables like '%read_only%';
+-----------------------+-------+
| Variable_name         | Value |
+-----------------------+-------+
| innodb_read_only      | OFF   |
| read_only             | OFF   |
| super_read_only       | OFF   |
| transaction_read_only | OFF   |
+-----------------------+-------+
4 rows in set (0.00 sec)
```

in the above all is off so we need to on for temporary if on permanent so wee need to change into the my.cnf file & server required restart for now we have set it to the temporary using Global variable.

Set Global read_only=ON # this will set to the read only except root user means in this root user can read write

Set Global super_read_only=ON # in this the root user can also not write or insert only read the data

```
mysql> set Global read_only=ON;
Query OK, 0 rows affected (0.00 sec)

mysql> set Global super_read_only=ON;
Query OK, 0 rows affected (0.00 sec)
```

While inserting data into slave server which is already in read only mode give the below pop or erro

```
mysql> insert into employee (id , name ) values(1,'MySQL DBA');
ERROR 1290 (HY000): The MySQL server is running with the --super-read-only option so it cannot execute this statement
mysql>
```

Common Replication Problems

❌ 1. Replication Stopped

Slave_SQL_Running: No

Cause:
Duplicate key
Missing table
Corrupt data

Check thread: run on master
SHOW PROCESSLIST

Look for:
Binlog Dump
Slave IO
Slave SQL

```
mysql> show processlist;
+----+-----------------+-------------+-----------+------------------+------+-----------------------------------------------------------------------+
| Id | User            | Host        | db        | Command          | Time | State                                                                 |
+----+-----------------+-------------+-----------+------------------+------+-----------------------------------------------------------------------+
|  7 | event_scheduler | localhost   | NULL      | Daemon           | 4328 | Waiting on empty queue                                                |
| 18 | repl_user       | db-02:37464 | NULL      | Binlog Dump GTID |  868 | Source has sent all binlog to replica; waiting for more updates       |
| 20 | root            | localhost   | replicadb | Query            |    0 | init                                                                  |
+----+-----------------+-------------+-----------+------------------+------+-----------------------------------------------------------------------+
```

❌ 2. Replication Lag

Reasons:
⦸ Slow disk
⦸ Big transactions
⦸ Low memory
⦸ Network delay

❌ 3. Data Mismatch

Caused by:
⦸ Manual writes on slave
⦸ Skipped errors
⦸ Bad restore
Never write on slave **!**

Fix Replication Errors (Basic)

⚠ Use only in emergency.

Skip one error:

```
STOP SLAVE;
SET GLOBAL SQL_SLAVE_SKIP_COUNTER=1;
START SLAVE;
```

(Not recommended long-term ❌)

Better → Rebuild replica ✓

High Availability (HA) Basics

Replication ≠ Full HA

HA = Auto failover.

Basic HA Setup:

```
Master ↔ Slave
↓
Failover Script / Tool
↓
VIP / Proxy
```

◈ Common HA Tools

Tool	Purpose
MHA	Automatic failover
Orchestrator	Topology manager
ProxySQL	Traffic routing

(Advanced topics — later weeks)

Production Best Practices

✓ **Do This**

✓ Use GTID

✓ Enable read_only on slave

✓ Monitor lag

✓ Daily health check

✓ Regular rebuild test

❌ **Avoid This**

⃠ Writing on slave

⃠ Skipping many errors

⃠ No monitoring

⃠ Same disk backup

⃠ No documentation

Testing for knowlege ?

Q1: How replication works ?
Ans :Binlog → Relay → Apply.

Q2: GTID advantage ?
Ans : Auto positioning.

Q3: How check health ?
Ans : SHOW SLAVE STATUS.

Q4: Lag reasons ?
Ans : Disk, network, big TX.

Q5: Replication vs HA ?
Ans : Copy vs auto failover.