MYSQL REPLICATION ERROR HANDLE/Standby Slave

Contents :

1. What we do when slave server already inserted the data or that id no already are present ?

2.We can set the slave server read only

Execute the following command on the slave server:
Mysql > SET GLOBAL read_only = 1;

Also changes into the my.cnf file

[mysqld]
read_only = 1 -- this is applicable but the super user are perform DML operation
super_read_only = 1 -- this is use the super user also not perform DML operation
After making changes restart the mysql server

sudo systemctl restart mysql

Note : Important Considerations

- **SUPER Privilege**: Users with the `SUPER` privilege can still make changes to the database, even when it is set to read-only. Be cautious with user privileges.

- **Replication Threads**: Replication threads are not affected by the read-only setting, so they can continue to apply changes from the master server.

- **Temporary Changes**: If you set the `read_only` variable dynamically (at runtime), it will not persist across MySQL server restarts. To make it persistent, ensure it is set in the MySQL configuration file.

You can skip the problematic event using sql_slave_skip_counter. This is a temporary fix and should be done with caution as it may lead to data inconsistency.

Mysql > STOP SLAVE;
Mysql > SET GLOBAL sql_slave_skip_counter = 1;
Mysql > START SLAVE;

This will skip the next event and allow the replication to continue. Check the slave status to ensure that replication has resumed:

Mysql > SHOW SLAVE STATUS\G;

If Necessary, Re-sync the Slave

If the data inconsistency cannot be resolved by the above methods, you may need to re-sync the slave with the master. This involves taking a fresh backup from the master and restoring it on the slave. Here's a high-level overview of the steps:

On the Master:

Lock the Tables (to ensure a consistent backup):

Mysql > FLUSH TABLES WITH READ LOCK;

Take a Backup:

mysqldump -u root -p --all-databases --master-data > master_backup.sql

Unlock the Tables:

Mysql > UNLOCK TABLES;

On the Slave:

Mysql > **Stop the Slave**:

Restore the Backup:

mysql -u root -p < master_backup.sql

Configure the Slave to Point to the Master (using the master_log_file and master_log_pos from the backup):

Mysql > CHANGE MASTER TO

MASTER_HOST='master_host',

MASTER_USER='replication_user',

MASTER_PASSWORD='replication_password',

MASTER_LOG_FILE='mysql-bin.000004',

MASTER_LOG_POS=3673;

Start the Slave:

Mysql > START SLAVE;

Check the Slave Status:

Mysql > SHOW SLAVE STATUS\G;

Ajit Yadav(DBA)