Implementation of Master and 2 Slave Node Replication at Radico Rampur

Table of Contents

Infrastructure Used

- Mysql 8.0.12 on Windows 2012 (IP 172.30.3.155) as **MASTER** Node .
- Mysql 8.0.12 on Windows 2012 (IP 172.30.3.220) as **Ist SLAVE Node**
- Mysql 8.0.12 on Linux / Centos 6.10 (IP 172.30.3.6) as **2nd SLAVE Node**

Implementation Process

Critically the objective was to **implementing without any down time**

Step 1 – basic requirement and configuration

1. Unique server_id for all the mysql servers in my.ini or my.cnf
    1. for 172.30.3.155 , server_id = 1
    2. for 172.30.3.220 , server_id = 2
    3. for 172.30.3.6 , server_id = 3

Ensure bin-log is activated on all server

Step 2 – Taking Dump and restore

Take persistent mysqldump from the master with binlog name and position

mysqldump -h 172.30.3.155 -uroot -proot --skip-lock-tables --single-transaction --flush-logs --master-data=2 -A > mysqldump.sql

Reference Document : https://linuxscriptshub.com/mysql-replication-setup-without-downtime/

Restore the Dump taken from the Master to slave Server

1. On Windows Server , Use Mysql Work Bench GUI Menu of data restore to restore the data
2. On Linux Server , use *mysql -uroot -p < mysqldump.sql* to restore the database .

Step 3 – making master and slave ready for replication

- Create replication privileged mysql user dedicated for replication purpose
    1. On Master
    2. CREATE USER 'slave_user'@'%';
    3. ALTER USER 'slave_user'@'%' IDENTIFIED BY 'Passw0rd';
    4. GRANT REPLICATION SLAVE ON *.* TO 'slave_user'@'%';

Step 4 – Enable Replication

1. Check the MASTER_LOG_FILE and MATER_LOG_POS. This information is available in the mysqldump.sql file from the master on slave
    1. *head mysqldump.sql -n80 | grep "MASTER_LOG_POS"*
    2. The Output will be “*-- CHANGE MASTER TO MASTER_LOG_FILE='binlog.000072', MASTER_LOG_POS=155;”*
2. Login into MySQL CLI or Shell and execute the below command to start replication
    1. *STOP SLAVE;*
    2. *CHANGE MASTER TO MASTER_HOST='172.30.3.155',MASTER_USER='slave_user',MASTER_PASSWORD='Passw0rd', MASTER_LOG_FILE='binlog.000072', MASTER_LOG_POS=155;*
    3. *START SLAVE;*

Step 5 – Replication Status check

On Slave Servers at mysql cli

SHOW SLAVE STATUS\G

Should Give Output without errors .

```
mysql> show slave status\G
*************************** 1. row ***************************
               Slave_IO_State: Waiting for master to send event
                  Master_Host: 172.30.3.155
                  Master_User: slave_user
                  Master_Port: 3306
                Connect_Retry: 60
              Master_Log_File: binlog.000073
          Read_Master_Log_Pos: 127606919
               Relay_Log_File: mysql-relay-bin.000007
                Relay_Log_Pos: 12171808
        Relay_Master_Log_File: binlog.000073
             Slave_IO_Running: Yes
            Slave_SQL_Running: Yes
              Replicate_Do_DB:
          Replicate_Ignore_DB:
           Replicate_Do_Table:
       Replicate_Ignore_Table:
      Replicate_Wild_Do_Table:
  Replicate_Wild_Ignore_Table:
                   Last_Errno: 0
                   Last_Error:
                 Skip_Counter: 0
          Exec_Master_Log_Pos: 127606919
              Relay_Log_Space: 12172016
              Until_Condition: None
               Until_Log_File:
                Until_Log_Pos: 0
           Master_SSL_Allowed: No
           Master_SSL_CA_File:
           Master_SSL_CA_Path:
              Master_SSL_Cert:
            Master_SSL_Cipher:
               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:
  Replicate_Ignore_Server_Ids:
             Master_Server_Id: 1
                  Master_UUID: bf9992ff-6c9b-11e9-8f14-30e1716afbdd
             Master_Info_File: mysql.slave_master_info
                    SQL_Delay: 0
          SQL_Remaining_Delay: NULL
      Slave_SQL_Running_State: Slave has read all relay log; waiting for more updates
           Master_Retry_Count: 86400
                  Master_Bind:
```

Last_IO_Error_Timestamp:
Last_SQL_Error_Timestamp:
Master_SSL_Crl:
Master_SSL_Crlpath:
Retrieved_Gtid_Set:
Executed_Gtid_Set:
Auto_Position: 0
Replicate_Rewrite_DB:
Channel_Name:
Master_TLS_Version:
Master_public_key_path:
Get_master_public_key: 0
1 row in set (0.00 sec)

Initially the Seconds_Behind_Master will be high value but it will reduce and should become 0 after few hours

my.cnf for Linux / centos mysql server

[mysqld]

datadir=/var/lib/mysql

socket=/var/lib/mysql/mysql.sock

log-error=/var/log/mysqld.log

pid-file=/var/run/mysqld/mysqld.pid

innodb-flush-method = O_DIRECT

innodb-flush-log-at-trx-commit = 2

innodb-file-per-table = 1

innodb-buffer-pool-size = 10G

default-storage-engine = InnoDB

 # CACHES AND LIMITS #

tmp-table-size = 1024M

max-heap-table-size = 32M

max-connections = 500

thread-cache-size = 500

open-files-limit = 65535

table-definition-cache = 4096

table-open-cache = 2048

server-id=3

relay-log = mysql-relay-bin