MYSQL Master/Slave Replication Setup

Version History

#	Version	Author	Reviewer
1			

Overview

The Proof of Concept (PoC) for setting up a MySQL Replication Cluster involves demonstrating the feasibility and functionality of implementing a replicated database environment. The PoC aims to showcase the benefits and capabilities of a MySQL Replication Cluster in a controlled environment.

During the PoC, a master-slave replication configuration is established, with a master node receiving write operations and propagating them to one or more slave nodes. The setup demonstrates features such as high availability, fault tolerance, load balancing, and scalability.

Pre-requisites

Installations required to perform the POC.

1. Minimum 2 Nodes – one master & one slave. **[amazon-linux/ ubuntu]**
2. Connectivity to install packages using yum or through corp proxy if any.
3. Install **Mysql** on both Master and slave nodes
4. Root access to the servers.

5. Each server's firewall should allow traffic on ports **22** and **Mysql port 3306**.

Procedure

To perform this POC I'm going to use two node cluster one master & one slave and launch two ubuntu 22.04 servers and give it as a name for that servers Master and Slave

Step1: Installation on MySQL server on Master Node

First, we must install MySQL server on Master for that installation run following below commands

- sudo apt update
- sudo apt upgrade
- sudo apt install mysql-server
- sudo systemctl status mysql

Step2: Configure MySQL Master Node

1. First, we will configure the master node with the required configuration. Go to this path
   sudo vim /etc/mysql/mysql.conf.d/mysqld.cnf
2. In the same file, find the line containing **bind-address = 127.0.0.1** and replace that IP address with the **private IP address** of your **master replication server**. So, the line will look like: bind-address = 172.31.91.71
3. Next, find the following lines in the file: **# Server-id =1**, lines have been commented, just uncomment the line (remove that # symbol in front of server-id =1).
4. Paste below script in that configuration file

   log_bin = /var/log/mysql/mysql-bin.log
   log_bin_index =/var/log/mysql/mysql-bin.log.index
   relay_log = /var/log/mysql/mysql-relay-bin
   relay_log_index = /var/log/mysql/mysql-
5. After changes Restart the file to run this command **"sudo systemctl restart mysql"**
6. Login to MySQL server as the root user, switch to root user (**sudo su root**) and run this command (**sudo mysql -u root -p**)
7. Create a user named **demo1** with a strong password by using the command below.
8. mysql > CREATE USER 'demo1'@'172.31.16.177' IDENTIFIED WITH mysql_native_password BY 'Demo@123';
9. This user will be used by the slaves to replicate the data from the master. Replace "172.31.16.177" with your **slave-node private Ip**
10. mysql > GRANT REPLICATION SLAVE ON *.* TO 'demo1'@'172.31.27.16'; (Replace '172.31.27.16' with **slave node private ip**)
11. mysql > FLUSH PRIVILEGES;
12. mysql> SHOW MASTER STATUS;

```
+------------------+----------+--------------+------------------+-------------------+
| File             | Position | Binlog_Do_DB | Binlog_Ignore_DB | Executed_Gtid_Set |
+------------------+----------+--------------+------------------+-------------------+
| mysql-bin.000002 |      557 |              |                  |                   |
+------------------+----------+--------------+------------------+-------------------+
1 row in set (0.00 sec)
```

13. Note it down **MASTER_LOG_FILE** (mysql-bin.000002) and **MASTER_LOG_POS** (557)

Step3: Installation on MySQL Server on Slave Node

1. Same thing we have to do hear also, what you did in the step-1 for installing MySQL server on master Node
2. Go to this path **sudo vim /etc/mysql/mysql.conf.d/mysqld.cnf** in the configuration file, find the line containing **bind-address = 127.0.0.1** and replace that IP address with the **private IP** address of your **slave replication server**. So, the line will look like: bind-address = 172.31.91.71
3. Next, find the following lines in the file: **# Server-id =1**, lines have been commented, just uncomment the line (remove that # symbol in front of server-id =1) and change **server-id =2**.
4. Paste below script in that configuration file

   log_bin = /var/log/mysql/mysql-bin.log
   log_bin_index =/var/log/mysql/mysql-bin.log.index
   relay_log = /var/log/mysql/mysql-relay-bin
   relay_log_index = /var/log/mysql/mysql-
5. After changes Restart the file to run this command **"sudo systemctl restart mysql"**
6. switch to root user (**sudo su root**) and run this command (**sudo mysql -u root -p**) and run this below command
7. CHANGE MASTER TO MASTER_HOST ='172.31.27.200', MASTER_USER ='demo1', MASTER_PASSWORD ='Demo@123', MASTER_LOG_FILE = 'mysql-bin.000002', MASTER_LOG_POS = 853, SOURCE_SSL=1;
8. Hear we need to change **'172.31.27.200'** with **master node private ip** and Replace **MASTER_LOG_FILE** & **MASTER_LOG_POS** with the values, you got from step 2 in master configuration.
9. Start the slave by using following command **START SLAVE;**
10. Check the replication status **SHOW REPLICA STATUS \G;**

Step4: Test MySQL Master-Slave Replication

To test if your MySQL master slave replication works, just create a database in your master server and see if it is replicated in the slave server. If you can see the database in the slave, then it is working fine.

1. On Master Server

- Login to master mySQL CLI.

  mysql –u root -p
- Create a database named testdb

  CREATE DATABASE Poornima;
- Run this command **show databases;**
- Example, output.

```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
```

| performance_schema |
| **Poornima** |
| sys |
+--------------------+
5 rows in set (0.00 sec)

2. **On Slave Server**

- Now login to your slave server mySQL CLI.
  mysql -uroot -p
- Run this command **show databases;**
- List the Databases. You should see the **Poornima** database created from the master server.

References

Setup MySQL Master Slave Replication: Step By Step Guide (devopscube.com)

How To Install MySQL 8 on Amazon Linux 2 – TecAdmin

MYSQL Master/Slave Replication Setup by Using Linux Server

How To Install MySQL 8 on Amazon Linux 2 – TecAdmin

First, we have to install two linux servers master and slave
Connect to the master and slave server and install mysql server by using following commands

```
sudo amazon-linux-extras install epel -y
sudo yum install https://dev.mysql.com/get/mysql80-community-release-el7-5.noarch.rpm
sudo yum install mysql-community-server
systemctl active mysqld
systemctl start mysqld
systemctl status mysqld
```

sudo cat /var/log/mysqld.log | grep "A temporary password" → password will be generated copy that password
sudo mysql_secure_installation → hear we need to paste that password

- Enter password for user root: **[Enter current root password]**
- New password: **[Enter a new root password]**
- Re-enter new password: **[Re-Enter the new root password]**
- Estimated strength of the password: 100
  Change the password for root ? ((Press y|Y for Yes, any other key for No) : **n**
- Remove anonymous users? (Press y|Y for Yes, any other key for No) : **y**
- Disallow root login remotely? (Press y|Y for Yes, any other key for No) : **y**
- Remove test database and access to it? (Press y|Y for Yes, any other key for No) : **y**
- Reload privilege tables now? (Press y|Y for Yes, any other key for No) : **y**

- All done!

Connect to Master Server
Change the configuration, for that we need to go that path **sudo vi /etc/my.cnf**
Add this bind-address and server-id under [mysqld]
[mysqld]
bind-address = 172.31.17.130 →master private ip
server-id = 1
save the file and restart **sudo systemctl restart mysqld**

Connect to MySQL in master by running below command
mysql -u root -p →hear we have to enter what you created the password to connect mysql and run this command
CREATE USER 'demo1'@'172.31.18.189' IDENTIFIED WITH mysql_native_password BY 'Demo@123'; → Replace **'172.31.18.189'** with your **slave node private ip**

GRANT REPLICATION SLAVE ON *.* TO 'demo1'@'172.31.27.16'; → Replace **'172.31.27.16'** with **slave node private ip**

FLUSH PRIVILEGES;
SHOW MASTER STATUS;
mysql> SHOW MASTER STATUS;

```
+---------------+----------+--------------+------------------+-------------------+
| File          | Position | Binlog_Do_DB | Binlog_Ignore_DB | Executed_Gtid_Set |
+---------------+----------+--------------+------------------+-------------------+
| binlog.000002 |      853 |              |                  |                   |
+---------------+----------+--------------+------------------+-------------------+
```

1 row in set (0.00 sec)
Note down file name (binlog.000002) and position (853)

Connect to Slave Server
Change the configuration, for that we need to go that path **sudo vi /etc/my.cnf**
Add this bind-address and server-id under [mysqld]
[mysqld]
bind-address = 172.31.17.130 →slave private ip
server-id = 2
save the file and restart **sudo systemctl restart mysqld**

Connect to MySQL in master
mysql -u root -p →hear we have to enter what you created the password to connect mysql and run this command
CHANGE MASTER TO MASTER_HOST ='172.31.27.200', MASTER_USER ='demo1', MASTER_PASSWORD ='Demo@123', MASTER_LOG_FILE = 'mysql-bin.000002', MASTER_LOG_POS = 853, SOURCE_SSL=1;

Hear we need to replace **'172.31.27.200'** with **master node private ip** and master-log-file name and master-log-pos

START SLAVE;
SHOW REPLICA STATUS \G;

Go to Master Server and connect and create a database
CREATE DATABASE testdb;
show databases;
Go to Slave server and run show databases;