**MySQL Master-Slave Replication Setup Documentation**

**Prerequisites:**

* Install MySQL Server (Version 8.0 used in this example).
* Access to command prompt with Administrator rights.
* Configured MySQL instances for both master and slave (in this example, ports 3306 for master and 3307 for slave).

## **Directory Structure:**

* **Master server data location**: C:/Program Files/MySQL/MySQL Server 8.0/data
* **Slave server data location**: C:/MySQL/MySQL\_Replica/data\_slave

**Note**: For the slave server, we will create a replica of the master server by copying the MySQL Server 8.0 directory and pasting it into another directory.

## **Copy MySQL Server for Slave Configuration**

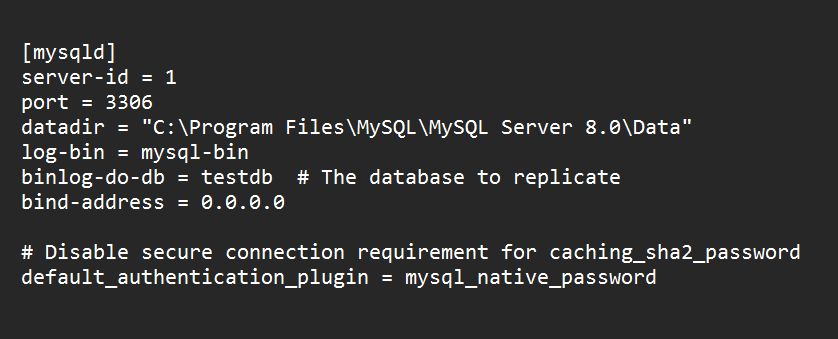
1. Navigate to C:/Program Files/MySQL/ and copy the MySQL Server 8.0 folder.
2. Paste it in the C:/MySQL/ directory and rename it to MySQL\_Replica.
3. This will act as the slave MySQL server.

## **Configuration of** my.ini **Files:**

### **Master Server (**my.ini**) Configuration**

The master server's configuration file (my.ini) is located at C:/Program Files/MySQL/MySQL Server 8.0/my.ini.

Edit the file to include the following:



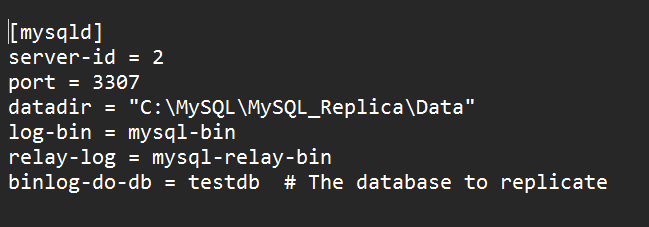
### **Explanation**:

* server-id = 1: This identifies the master server uniquely. Each server in a replication setup must have a unique server-id.
* port = 3306: Port number where the master MySQL server listens.
* log-bin = mysql-bin: Enables binary logging. This is necessary for replication as the slave will replicate the changes logged in the master's binary log.
* binlog-do-db = testdb: Specifies the database (testdb) that will be replicated to the slave.

### **Slave Server (**my.ini**) Configuration:**

The slave server's configuration file (my.ini) is located at C:/MySQL/MySQL\_Replica/my.ini.

Edit the file to include the following:



After Configured MySQL instances for both master and slave (in this example, ports 3306 for master and 3307 for slave).

**Step 1: Initialize the MySQL Replica Directory:**

Before setting up the slave server, we need to initialize the data directory for replication purposes.

1. Open **Command Prompt as Administrator**.
2. Run the following commands to initialize the replica:

C:\Windows\System32> cd C:\Program Files\MySQL\MySQL Server 8.0\bin

This command navigates to the MySQL installation's bin directory where MySQL binaries (e.g., mysqld) are located.

C:\Program Files\MySQL\MySQL Server 8.0\bin>mysqld --initialize --datadir="C:\MySQL\MySQL\_Replica\Data"

This command initializes a new data directory for the replica. The --datadir flag specifies the location where MySQL will store its data for the replica.

C:\Program Files\MySQL\MySQL Server 8.0\bin>cd C:\MySQL\MySQL\_Replica\bin

Navigate to the bin directory of your replica setup to manage the MySQL daemon for the replica.

C:\MySQL\MySQL\_Replica\bin>mysqld --defaults-file="C:\MySQL\MySQL\_Replica\my.ini"

Start the MySQL server for the replica with the configuration provided in my.ini.

### **Step 2: Initialize the Master Server**

Next, we need to configure the master server to allow replication.

1. Open another **Command Prompt as Administrator**.
2. Run the following commands to start the MySQL master server:

C:\Windows\System32>cd "C:\Program Files\MySQL\MySQL Server 8.0\bin"

Navigate to the master MySQL installation directory.

C:\Program Files\MySQL\MySQL Server 8.0\bin>mysqld --defaults-file="C:\Program Files\MySQL\MySQL Server 8.0\my.ini"

Start the MySQL master server using its configuration file my.ini.

**Step 3: Configure the Master Server for Replication**

Log in to the MySQL master server to create a replication user.

1. Run the following command to log into MySQL:

C:\Program Files\MySQL\MySQL Server 8.0\bin>mysql -u root -p -P 3306

After logging in, create a user for replication and grant necessary privileges:

CREATE USER 'replica\_user3'@'localhost' IDENTIFIED BY 'password';

Creates a new user replica\_user3 that the replica will use to authenticate with the master.

GRANT REPLICATION SLAVE ON \*.\* TO 'replica\_user3'@'localhost';

Grants the user replication privileges to allow the slave to replicate from the master.

FLUSH PRIVILEGES;

Refreshes MySQL privileges to apply the changes.

**Step 4: Get Master Status:**

To replicate data, you need the master log file name and log position.

Run the following command:

SHOW MASTER STATUS;

You will get an output like this:

+------------------+----------+--------------+------------------+-------------------+

| File | Position | Binlog\_Do\_DB | Binlog\_Ignore\_DB | Executed\_Gtid\_Set |

+------------------+----------+--------------+------------------+-------------------+

| mysql-bin.000004 | 2269 | testdb | | |

+------------------+----------+--------------+------------------+-------------------+

* **File**: Name of the binary log file.
* **Position**: The log position.

You will need this information to configure the slave.

### **Step 5: Configure the Slave Server**

1. Open another **Command Prompt as Administrator** and log in to the replica server:

C:\Windows\System32>mysql -u root -p -P 3307

Stop the slave process before configuring it: (if you tried previously)

STOP SLAVE;

Now, configure the slave to connect to the master:

CHANGE MASTER TO

MASTER\_HOST='localhost',

MASTER\_USER='replica\_user3',

MASTER\_PASSWORD='password',

MASTER\_PORT=3306,

MASTER\_LOG\_FILE='mysql-bin.000004',

MASTER\_LOG\_POS=2269;

**Explaination:**

 **MASTER\_HOST**: Hostname or IP of the master.

 **MASTER\_USER**: The replication user created on the master.

 **MASTER\_PASSWORD**: The replication user's password.

 **MASTER\_PORT**: The port on which the master is running (3306).

 **MASTER\_LOG\_FILE**: The binary log file from SHOW MASTER STATUS.

 **MASTER\_LOG\_POS**: The log position from SHOW MASTER STATUS.

Start the slave process:

START SLAVE;

**Step 6: Verify Slave Status:**

To check if the replication is running correctly, run the following command:

SHOW SLAVE STATUS\G;

Look for the following values in the output:

* **Slave\_IO\_Running: Yes**
* **Slave\_SQL\_Running: Yes**

If both are Yes, the replication is successfully established.

### **Step 7: Test the Replication**

Switch to the master database:

USE testdb;

# MySQL Replication: Useful Commands

1. **SHOW BINARY LOGS;**  
   Displays the list of binary log files on the master server.
2. **SHOW VARIABLES LIKE '%log\_bin%';**  
   Shows whether binary logging is enabled and related settings.
3. **SELECT User, Host FROM mysql.user;**  
   Lists all MySQL users and their associated hosts.
4. **DROP USER 'replication\_user'@'%';**  
   Deletes the specified replication user.
5. **SHOW VARIABLES LIKE 'server\_id';**  
   Displays the server's unique server\_id used in replication.
6. **SELECT @@datadir;**  
   Shows the data directory path where MySQL stores its data.
7. **SHOW SLAVE STATUS\G;**  
   Provides detailed information about the slave server's replication status.
8. **START SLAVE;**  
   Starts the replication process on the slave server.
9. **STOP SLAVE;**  
   Stops the replication process on the slave server.
10. **RESET SLAVE;**  
    Resets all replication settings on the slave server.
11. **SHOW MASTER STATUS;**  
    Displays the master server’s current binary log file and position.