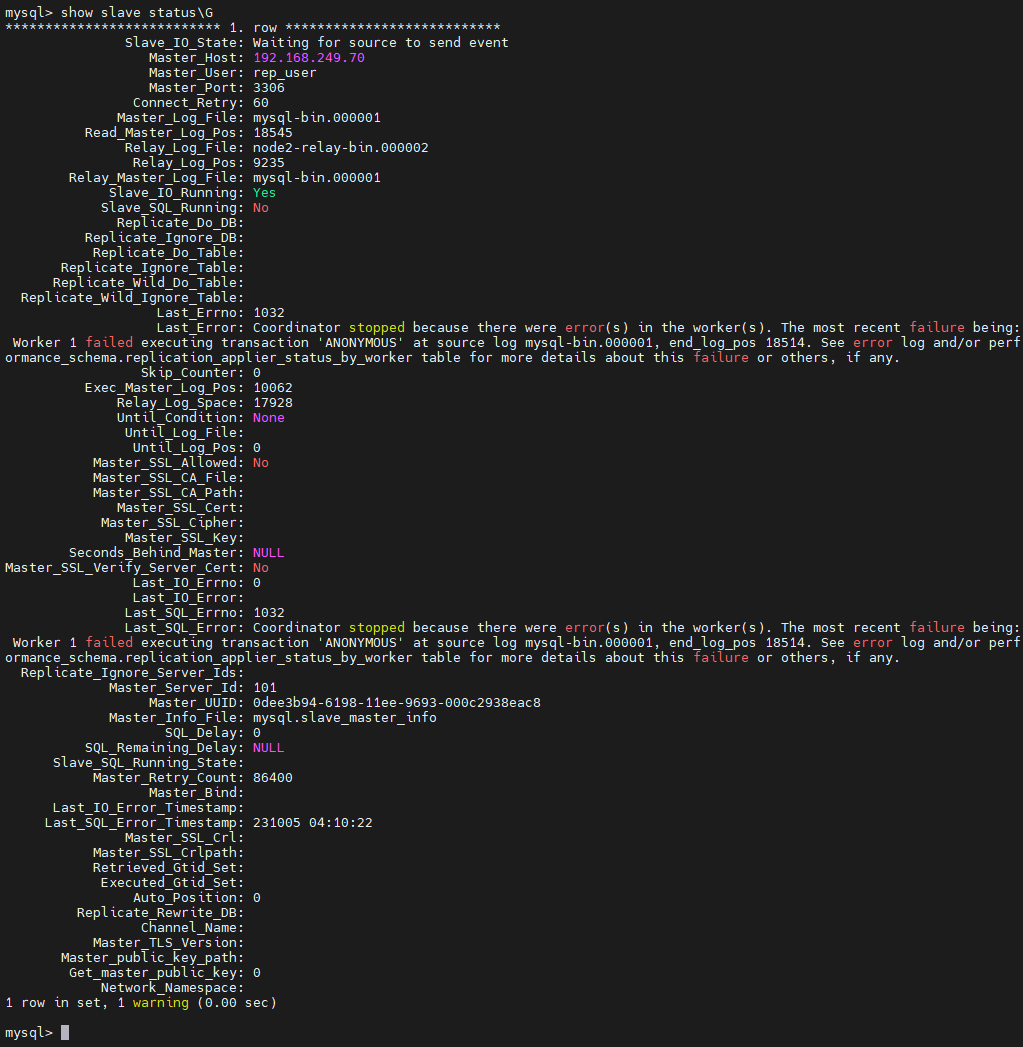
**Setup Master-Master replication in MySQL on Ubuntu**

*Lưu ý: Trước khi bạn bắt đầu, hãy đảm bảo rằng bạn đã cài đặt MySQL trên cả hai máy chủ và có quyền root trên cả hai máy.*

|  |  |
| --- | --- |
| Master 1 Server 1 | Master 2 Server 2 |
|  | |
| Step 1: Prerequisites | |
| 1.1 On Master Server 1 | **1.2 On Master Server 2** |
| $ ifconfig | $ ifconfig |
| $ sudo systemctl status mysql | $ sudo systemctl status mysql |
| ping [Đảm bảo hai server thông nhau] | $ ping [Đảm bảo hai server thông nhau] |
| Đảm bảo rằng cả hai máy chủ có thể giao tiếp với nhau qua mạng. Mở các cổng cần thiết (cổng mặc định của MySQL là 3306) và cho phép lưu lượng MySQL thông qua tường lửa. | |
| $ sudo ufw allow 3306/tcp | $ sudo ufw allow 3306/tcp |
|  | |
| Step 2: Configure Master DB Server on Server-1 | |
| 2.1 Open & Edit /etc/mysql/mysql.conf.d/mysqld.cnf file |  |
| Sau khi kiểm tra, thực hiện chạy các lệnh sau: |  |
| $ sudo nano /etc/mysql/mysql.conf.d/mysqld.cnf  Thêm cấu hình sau:  # Unique ID of the MySQL server.  # This ID can not be re-used in any nodes in the cluster.  server-id = 101  ###  # This is the file in which all the replication information is stored.  log-bin = /var/log/mysql/mysql-bin.log |  |
| Tiếp tục, kéo xuống dưới, comment hai dòng 37, 38 như hình: |  |
| Lưu lại và thoát khỏi trình soạn thảo. |  |
| 2.2. Restart MySQL server |  |
| $ sudo systemctl restart mysql |  |
| 2.3 Create a new user for the Replication service in the Master node |  |
| Tiếp theo, sau khi khỏi động lại dịch vụ mysql, chạy lệnh sau:  $ sudo mysql -u root -p |  |
| Trong mysql, thực hiện chạy các lệnh sau:  $ create user 'rep\_user'@'%' identified by 'rep\_user\_PWD';  $ alter user 'rep\_user'@'%' identified with mysql\_native\_password by 'rep\_user\_PWD';  $ grant replication slave on \*.\* to 'rep\_user'@'%';  $ flush privileges; |  |
| Truy vấn thông tin người dùng bằng cách sử dụng câu lệnh SELECT như sau:  $ SELECT User, Host FROM mysql.user;  Xoá người dùng trong mysql:  DELETE FROM mysql.user WHERE User='rep\_user' AND Host='%'; |  |
| $ show master status\G |  |
| Step 3: Configure Master DB Server on Server-2 | |
| Chuyển sang master server 2: | $ sudo nano /etc/mysql/mysql.conf.d/mysqld.cnf  **Thêm cấu hình sau:**  # Unique ID of the MySQL server.  # This ID can not be re-used in any nodes in the cluster.  server-id = 102  ###  # This is the file in which all the replication information is stored.  log-bin = /var/log/mysql/mysql-bin.log |
|  | Tiếp tục, kéo xuống dưới, comment hai dòng 37, 38 như hình: |
|  | Lưu lại và thoát khỏi trình soạn thảo. |
|  | $ sudo systemctl restart mysql |
|  | Tiếp theo, sau khi khỏi động lại dịch vụ mysql, chạy lệnh sau:  sudo mysql -u root -p |
|  | Trong mysql, thực hiện chạy các lệnh sau:  $ create user 'rep\_user'@'%' identified by 'rep\_user\_PWD';  $ alter user 'rep\_user'@'%' identified with mysql\_native\_password by 'rep\_user\_PWD';  $ grant replication slave on \*.\* to 'rep\_user'@'%';  $ flush privileges; |
|  | **$ show master status\G** |
| Step 4: Configure Replication on Server-2 | |
|  | Trong mysql, tiếp tục chạy lệnh sau:  $ stop slave; |
|  | Thực hiện chỉnh sửa cấu hình sau:  CHANGE MASTER TO MASTER\_HOST='IP\_of\_Master\_Server\_1',  MASTER\_USER='rep\_user',  MASTER\_PASSWORD='rep\_user\_PWD',  MASTER\_LOG\_FILE='Log\_on\_Master\_Server\_1',  MASTER\_LOG\_POS=Position\_of\_log\_file\_on\_Master\_Server\_1;  'IP\_of\_Master\_Server\_1': Thay cho em bằng ip của server 1.  Hai lệnh dưới anh lấy thông tin cho em ở hai trường: File và Possition ở lệnh **show master status\**G đã thực hiện ở server 1.  'Log\_on\_Master\_Server\_1'  Position\_of\_log\_file\_on\_Master\_Server\_1    Ví dụ: MASTER\_LOG\_FILE='mysql-bin.000001',  MASTER\_LOG\_POS=1150;  **Lưu ý: Giá trị này lấy ở server 1** |
|  | Sau khi đã chỉnh sửa cấu hình, anh copy toàn bộ và chạy trong mysql. |
|  | Tiếp tục, trong mysql chạy tiếp lệnh sau:  $ start slave; |
| Step 5: Configure Replication on Server-1 | |
| Trên server 1, rong mysql tiếp tục chạy lệnh sau:  $ stop slave; |  |
| Tương tự, chỉnh sửa cấu hình sau như vừa thực hiện trên server 2:  CHANGE MASTER TO MASTER\_HOST='IP\_of\_Master\_Server\_2',  MASTER\_USER=' rep\_user',  MASTER\_PASSWORD=' rep\_user\_PWD',  MASTER\_LOG\_FILE='Log\_on\_Master\_Server\_2',  MASTER\_LOG\_POS=Position\_of\_log\_file\_on\_Master\_Server\_2;  Lưu ý: Giá trị MASTER\_LOG\_FILE, MASTER\_LOG\_POS lấy ở server 2. |  |
| Sau khi đã chỉnh sửa cấu hình, anh copy toàn bộ và chạy trong mysql. |  |
| $ start slave; |  |
| Chạy lại lệnh sau:  $ show slave status\G    Kết quả trả về như này là ok.  Lưu ý: Hai trường Slave\_IO\_Running, Slave\_SQL\_Running giá trị đều phải là Yes |  |
|  | **Tương tự, kiểm tra trạng thái với server 2:**  $ show slave status\G |
| Thực hiện kiểm tra kết quả: | |
| Trên server 1, trong mysql tạo một database test; |  |
|  | Trên server hai, kiểm tra xem database 2 đã tồn tại hay chưa: |
|  | Kịch bản tương tự thực hiện trên server 2. Nếu hai cơ sở dữ liệu đồng bộ được với nhau, vậy ta đã thành công. |
|  |  |
| Sử dụng lệnh SQL sau để kiểm tra trạng thái của các worker đang thực hiện sao lưu:  $ SELECT \* FROM performance\_schema.replication\_applier\_status\_by\_worker; | |



**How To Configure MySQL Group Replication on Ubuntu 20.04**