

# Mysql Master-Slave Auto Configuration Script

- [Script](#)

# Script

**Replication Script**





```

#!/bin/bash
# Function to execute SSH commands using sshpass
function execute_ssh_command {
    sshpass -p "$1" ssh -o StrictHostKeyChecking=no -p "$2" root@"$3" "$4"
}
# Prompt for SSH port, password, MySQL root password, and server IP for master and slave servers
read -p "Enter SSH port for the master server: " ssh_port_master
read -s -p "Enter password for root user on the master server: " ssh_password_master
read -s -p "Enter MySQL root password for the master server: " mysql_root_pass_master
read -p "Enter master server IP address: " server_ip_master
echo
read -p "Enter SSH port for the slave server: " ssh_port_slave
read -s -p "Enter password for root user on the slave server: " ssh_password_slave
read -s -p "Enter MySQL root password for the master server: " mysql_root_pass_slave
read -p "Enter slave server IP address: " server_ip_slave
# Configuration variables for master
bind_address_master="$server_ip_master"
server_id_master="1"
log_bin="mysql-bin"
# SSH command block for master server
ssh_command_master="
    # Function to check if a line exists in a file
    function check_line_in_file {
        grep -qF \"\$1\" \"\$2\"
    }
    # Check and append configurations in mysql-server.cnf
    if ! check_line_in_file \"bind-address = $bind_address_master\" /etc/my.cnf.d/mysql-server.cnf; then
        echo \"bind-address = $bind_address_master\" >> /etc/my.cnf.d/mysql-server.cnf
    fi
    if ! check_line_in_file \"server-id = $server_id_master\" /etc/my.cnf.d/mysql-server.cnf; then
        echo \"server-id = $server_id_master\" >> /etc/my.cnf.d/mysql-server.cnf
    fi
    if ! check_line_in_file \"log_bin = $log_bin\" /etc/my.cnf.d/mysql-server.cnf; then
        echo \"log_bin = $log_bin\" >> /etc/my.cnf.d/mysql-server.cnf
    fi
    # Restart MySQL service
    systemctl restart mysqld
    # Check if the replica_user exists in MySQL
    if ! mysql -u root -p'$mysql_root_pass_master' -e \"SELECT User FROM mysql.user WHERE
User='replica_user'\" | grep -q 'replica_user'; then
        mysql -u root -p'$mysql_root_pass_master' -e \"
            CREATE USER 'replica_user'@'%' IDENTIFIED WITH mysql_native_password BY
'$mysql_root_pass_master';
            GRANT REPLICATION SLAVE ON *.* TO 'replica_user'@'%';
            FLUSH PRIVILEGES;
        \"
        echo 'MySQL user replica_user created.'
    else
        echo 'MySQL user replica_user already exists. Skipping user creation.'
    fi
    # Retrieve and print master status from MySQL
    mysql_output_master=$(mysql -u root -p'$mysql_root_pass_master' -e \"SHOW MASTER STATUS\\G\")
    echo \"MySQL Output:\"
    echo \"\$mysql_output_master\"
    # Parse and print 'File' and 'Position' from master status
    mysql_file=$(echo \"\$mysql_output_master\" | awk '/File:/ {print \$2}')
    mysql_position=$(echo \"\$mysql_output_master\" | awk '/Position:/ {print \$2}')
    echo \"MySQL File: \$mysql_file\"
    echo \"MySQL Position: \$mysql_position\"
    # Save the parsed file and position to a temporary file for later use
    echo \$mysql_file > /tmp/mysql_file.txt
    echo \$mysql_position > /tmp/mysql_position.txt

```

## Script Explanation

- This function uses `sshpass` to execute SSH commands on a remote server. The function takes four arguments: the SSH password, the SSH port, the server IP, and the command to execute.

```
function execute_ssh_command {  
    sshpass -p "$1" ssh -o StrictHostKeyChecking=no -p "$2" root@"$3" "$4"  
}
```

- These prompts gather necessary details from the user to configure both the master and slave servers, such as SSH ports, passwords, and IP addresses.

```
read -p "Enter SSH port for the master server: " ssh_port_master  
read -s -p "Enter password for root user on the master server: " ssh_password_master  
read -s -p "Enter MySQL root password for the master server: " mysql_root_pass_master  
read -p "Enter master server IP address: " server_ip_master  
echo  
read -p "Enter SSH port for the slave server: " ssh_port_slave  
read -s -p "Enter password for root user on the slave server: " ssh_password_slave  
read -s -p "Enter MySQL root password for the master server: " mysql_root_pass_slave  
read -p "Enter slave server IP address: " server_ip_slave
```

- These variables define the master server's configuration settings.

```
bind_address_master="$server_ip_master"  
server_id_master="1"  
log_bin="mysql-bin"
```

- SSH block for master

```

ssh_command_master="
# Function to check if a line exists in a file
function check_line_in_file {
    grep -qF \"\$1\" \"\$2\"
}
# Check and append configurations in mysql-server.cnf
if ! check_line_in_file \"bind-address = \$bind_address_master\" /etc/my.cnf.d/mysql-server.cnf;
then
    echo \"bind-address = \$bind_address_master\" >> /etc/my.cnf.d/mysql-server.cnf
fi
if ! check_line_in_file \"server-id = \$server_id_master\" /etc/my.cnf.d/mysql-server.cnf; then
    echo \"server-id = \$server_id_master\" >> /etc/my.cnf.d/mysql-server.cnf
fi
if ! check_line_in_file \"log_bin = \$log_bin\" /etc/my.cnf.d/mysql-server.cnf; then
    echo \"log_bin = \$log_bin\" >> /etc/my.cnf.d/mysql-server.cnf
fi
# Restart MySQL service
systemctl restart mysqld
# Check if the replica_user exists in MySQL
if ! mysql -u root -p'$mysql_root_pass_master' -e \"SELECT User FROM mysql.user WHERE
User='replica_user'\" | grep -q 'replica_user'; then
    mysql -u root -p'$mysql_root_pass_master' -e \"
        CREATE USER 'replica_user'@'%' IDENTIFIED WITH mysql_native_password BY
'$mysql_root_pass_master';
        GRANT REPLICATION SLAVE ON *.* TO 'replica_user'@'%';
        FLUSH PRIVILEGES;
    \"
    echo 'MySQL user replica_user created.'
else
    echo 'MySQL user replica_user already exists. Skipping user creation.'
fi
# Retrieve and print master status from MySQL
mysql_output_master=\$(mysql -u root -p'$mysql_root_pass_master' -e \"SHOW MASTER
STATUS\\G\")
echo \"MySQL Output:\"
echo \"\\$mysql_output_master\"
# Parse and print 'File' and 'Position' from master status
mysql_file=\$(echo \"\\$mysql_output_master\" | awk '/File:/ {print \$2}')
mysql_position=\$(echo \"\\$mysql_output_master\" | awk '/Position:/ {print \$2}')
echo \"MySQL File: \\$mysql_file\"
echo \"MySQL Position: \\$mysql_position\"
# Save the parsed file and position to a temporary file for later use
echo \\$mysql_file > /tmp/mysql_file.txt
echo \\$mysql_position > /tmp/mysql_position.txt
# Exit the SSH session
exit
\"

```

This block does the following on the master server:

1. Defines a function to check if a line exists in a file.
2. Configures mysql-server.cn with the master server settings if they are not already present.
3. Restarts the MySQL service.
4. Creates a replica\_user for replication if it doesn't already exist.

5. Retrieves the current master status, including the File and Position.
  6. Saves the File and Position values to temporary files.
- This line executes the defined command block on the master server.

```
execute_ssh_command "$ssh_password_master" "$ssh_port_master" "$server_ip_master"  
"$ssh_command_master"
```

- This section retrieves the File and Position values from the master server, which are necessary for setting up the slave server.

```
mysql_file=$(sshpass -p "$ssh_password_master" ssh -o StrictHostKeyChecking=no -p  
"$ssh_port_master" root@"$server_ip_master" "cat /tmp/mysql_file.txt")  
mysql_position=$(sshpass -p "$ssh_password_master" ssh -o StrictHostKeyChecking=no -p  
"$ssh_port_master" root@"$server_ip_master" "cat /tmp/mysql_position.txt")  
echo "MySQL configuration and user setup on master server completed."  
echo "MySQL File: $mysql_file"  
echo "MySQL Position: $mysql_position"
```

- These variables define the slave server's configuration settings.

```

ssh_command_slave="
# Function to check if a line exists in a file
function check_line_in_file {
    grep -qF \"\$1\" \"\$2\"
}
# Check and append configurations in mysql-server.cnf
if ! check_line_in_file \"bind-address = $bind_address_slave\" /etc/my.cnf.d/mysql-server.cnf;
then
    echo \"bind-address = $bind_address_slave\" >> /etc/my.cnf.d/mysql-server.cnf
fi
if ! check_line_in_file \"server-id = $server_id_slave\" /etc/my.cnf.d/mysql-server.cnf; then
    echo \"server-id = $server_id_slave\" >> /etc/my.cnf.d/mysql-server.cnf
fi
if ! check_line_in_file \"log_bin = $log_bin\" /etc/my.cnf.d/mysql-server.cnf; then
    echo \"log_bin = $log_bin\" >> /etc/my.cnf.d/mysql-server.cnf
fi
# Restart MySQL service
systemctl restart mysqld
# Access MySQL and execute replication setup queries
mysql -u root -p'$mysql_root_pass_slave' -e \"
    STOP SLAVE;
    RESET SLAVE;
    CHANGE MASTER TO MASTER_HOST='$server_ip_master', MASTER_USER='replica_user',
MASTER_PASSWORD='$mysql_root_pass_master', MASTER_LOG_FILE='$mysql_file',
MASTER_LOG_POS=$mysql_position;
    START SLAVE;
    SHOW SLAVE STATUS\G;
\" > /tmp/slave_status_output.txt
# Print the output of SHOW SLAVE STATUS
echo 'Slave Status Output:'
cat /tmp/slave_status_output.txt
# Exit the SSH session
exit
"

```

This block does the following on the slave server:

1. Defines a function to check if a line exists in a file.
  2. Configures mysql-server.cnf with the slave server settings if they are not already present.
  3. Restarts the MySQL service.
  4. Configures the slave to replicate from the master using the retrieved File and Position values.
  5. Retrieves and prints the slave status.
- This line executes the defined command block on the slave server.

```

execute_ssh_command "$ssh_password_slave" "$ssh_port_slave" "$server_ip_slave"
"$ssh_command_slave"

```

- This line indicates that the MySQL replication setup on the slave server is complete.

echo "MySQL replication setup on slave server completed."

.....Thank  
you.....