SETUP A MASTER-TO-MASTER REPLICATION BETWEEN TWO MARIADB SERVERS

TAGS: SQL REPLICATION DATABASE MYSQL MARIADB CENTOS7

Master-to-Master Replication Between Two MariaDB Servers

This tutorial will have you get up and running an sql master-to-master replication. If you don't yet have any idea of what it means, a master-to-master is a type of replication process which allows data to be stored by a group of computers/servers that can be updated by any member of a group.

In this example we will group 2 servers together and replicate data to each other making each server a master and a slave database server at the same time.

PREREQUISITES

For this tutorial we shall assume that we have the following already set-up and ready to be re-configured:
- 2 servers with pre-installed MariaDB server (If you haven't done this yet there is a tutorial in this website (/out?u=https%3A%2F%2Fwww.vpsserver.com%2Fcommunity%2Ftutorials%2F8%2Finstalling-lamp-linux-apache-mysql-and-php-stack-on-centos-7-64bit%2F) that you can follow)
- A sample database
- Root Access to the server
- An SSH client (You can download putty (/out?u=http%3A%2F%2Fwww.chiark.greenend.org.uk%2F~sgtatham%2Fputty%2Fdownload.html) or bitvise (/out?u=https%3A%2F%2Fwww.bitvise.com%2Fssh-client-download) depends on your operating system and liking)

The servers that I have has the following information. You will have a different server ip but the process is the same so please do not be confused.

```
server_a (192.168.0.12)
server_b (192.168.0.15)
```

The sample database name shall be replicate.db. This database does not contain any data for now.

When you have all of the above ready, we can now start setting up our master-to-master replication. ___

CONFIGURING SERVER_A

This shall be our first master and we will need to edit the configuration files to enable replication. We will do.

```
sudo nano /etc/my.cnf
```

And under the

```
[mysqld]
```

section you have to add the following information.

```
log-bin
server_id=1
replicate-do-db=replicate
bind-address=192.168.0.12
```

Next, restart our sql server.

```
systemctl restart mariadb
```

Login to the MariaDB server as root user and create a master user with necessary grants. Please note that you have to supply the following details for *master_username* & *master_password*.

```
CREATE USER '$master_username'@'%' IDENTIFIED BY '$master_password';



GRANT REPLICATION SLAVE ON *.* TO '$master_username'@'%';



FLUSH PRIVILEGES;



SHOW MASTER STATUS;
```

The command 'SHOW MASTER STATUS' shall return to you the current binary log which is the exact location which indicates where the other master should start replication from.

```
+------------------+----------+--------------+------------------+
| File             | Position | Binlog_Do_DB | Binlog_Ignore_DB|
+------------------+----------+--------------+------------------+
| mariadb-bin.000001 |      626 |            |                  |
+------------------+----------+--------------+------------------+
1 row in set (0.00 sec)
```

Please take note of the position and file. You will need this information later on. ___

CONFIGURING SERVER_B

Make sure to copy the same sample database from server_a to this server before continuing with replication. This shall be our second master and we will need to edit the configuration files to enable replication. We will do.

```
sudo nano /etc/my.cnf
```

And under the

```
[mysqld]
```

section you have to add the following information.

```
log-bin

server_id=2

replicate-do-db=replicate

bind-address=192.168.0.15
```

Next, restart our sql server.

```
systemctl restart mariadb
```

Login to the MariaDB server as root user and create a master user with necessary grants. Please note that you have to supply the following details for *master_username* & *master_password*.

```
CREATE USER '$master_username'@'%' IDENTIFIED BY '$master_password';
```

```
GRANT REPLICATION SLAVE ON *.* TO '$master_username'@'%';
```

```
FLUSH PRIVILEGES;
```

The nest step involves taking the information we have from server_a and putting these information into this server. the following should be typed.

```
SLAVE STOP;
```

```
CHANGE MASTER TO MASTER_HOST = '192.168.0.12', MASTER_USER = '$master_username', MASTER_PASSWORD = '$master_password',
MASTER_LOG_FILE = 'mariadb-bin.000001', MASTER_LOG_POS = 626;
```

```
SLAVE START;
```

The last thing we should do on server_b to complete the replication is to take the information on the master log file and position so we can enter it on server_a.

```
SHOW MASTER STATUS;
```

The command 'SHOW MASTER STATUS' shall return to you the current binary log which is the exact location which indicates where the other master should start replication from.

```
+------------------+----------+--------------+------------------+
| File             | Position | Binlog_Do_DB | Binlog_Ignore_DB|
+------------------+----------+--------------+------------------+
| mariadb-bin.000002 |      635 |            |                  |
+------------------+----------+--------------+------------------+
1 row in set (0.00 sec)
```

Please take note of the position and file. You will need this information on the final step. ___

COMPLETING THE REPLICATION ON SERVER_A

We will need to finish the replication process for server_a so we have to login to the mariadb server and enter the information we just got from server_b.

```
SLAVE STOP;
```

```
CHANGE MASTER TO MASTER_HOST = '192.168.0.15', MASTER_USER = '$master_username', MASTER_PASSWORD = '$master_password',
MASTER_LOG_FILE = 'mariadb-bin.000002', MASTER_LOG_POS = 635;
```

```
SLAVE START;
```

TESTING THE MASTER-TO-MASTER REPLICATION

Now that all configuration is done we will need to test if our setup is working. While inside server_a, login to the MariaDB server and create a new table in replicate.db.

```
CREATE TABLE replicate.sample (`name` varchar(18));
```

Login to server_b, you should be able to see the new 'sample' table on the replicate database.

```
SHOW TABLES IN replicate;
```

You should see the following information:

```
+---------------------+
| Tables_in_replicate |
+---------------------+
| sample              |
+---------------------+
1 row in set (0.00 sec)
```

To check that replication is working from server_b to server_a is to do the same sql commands and check if the table exists in the other server. If it do, then our master-to-master replication is working as expected.

There you go! Now you can replicate your data to any server without worrying about downtime. This should help administrators wherein they want to have a fail-safe system in place.

RELATED TUTORIALS

- Install Percona Server 5.7 MySQL on Ubuntu, CentOS or Debian (/community/tutorials/3920/install-percona-server-5-7-mysql-ubuntu-centos-debian/)
- Installing Drupal on Ubuntu 16 VPS Server (/community/tutorials/49/installing-drupal-on-ubuntu-16-vps-server/)
- CentOS compiler installed under the Apache, PHP, MySQL (/community/tutorials/3942/centos-compiler-installed-under-the-apache-php-mysql/)
- Install Node.js on CentOS 7 (/community/tutorials/20/install-node-js-on-centos-7/)
- Install and Configure CSF (ConfigServer Firewall) on Centos7 64bit (/community/tutorials/11/install-and-configure-csf-configserver-firewall-on-centos7-64bit/)

COMMENTS (1)

GERBREOWN REPLY

+1 8 MONTHS AGO

I just followed this post and it is NOT working for me.

A couple of errors I found were "SLAVE STOP/START" should be "STOP/START SLAVE". Also all of those "&#039"s made it hard to read when ' would have been much easier to read.

BTW I am using Ubuntu 17.10 and MariaDB 10.2.13

Thanks,

LEAVE COMMENT

Please choose login or register to comment in this thread.

Login Register

Email

Password

Token

A token is only required when you have set this up in your account

Message

Receive email notifications when my comment receives comments

I'm not a robot

reCAPTCHA
Privacy - Terms

SUBMIT