MySQL :How to Configure MySQL Master-Slave Replication on RHEL

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

MySQL replication procedure enabled you to automatically copy data from one MYSQL database server to one or more MYSQL servers.

MySQL supports a number of replication topologies with Master/Slave topology being one of the most well-known topologies in which

one database server acts as the master, while one or more servers act as slaves. By default, the replication is asynchronous where

the master sends events that describe database modifications to its binary log and slaves request the events when they are ready.

This sort of replication topology is good for deploying of read replicas for read scaling, live databases backup

for disaster recovery and for analytics jobs.

MySQL 8.0 supports different methods of replication. The traditional method is based on replicating events from the master's binary log,

and requires the log files and positions in them to be synchronized between master and slave.

Prerequisites

\*\*\*\*\*\*\*\*\*\*\*\*\*

Setup

\*\*\*\*\*\*\*

ROLE HOST OS MYSQL SERVER VERSION

Master 192.168.1.51(linux2) OEL 6.5 8.0.19

Slave 192.168.1.61(linux3) OEL 6.5 8.0.19

vaidate value of

show variables like '%uuid%';

Step 1: Configure the Master Server.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

-Let's configure the master MySQL server and perform below changes.

-enable MySQL server to listen on the private IP.

-Set a unique server ID.

-Enable the binary logging.

#

cp /etc/my.cnf /etc/my.cnf\_bkp\_5may2020

vi /etc/my.cnf

bind-address = 192.168.1.51

server-id = 1

log\_bin = mysql-bin #Hechanged this location to /mysql/logs/mysql-bin

-Once done, restart the MySQL service for changes to take effect

# service mysqld restart

[root@linux2 ~]#

mysql -u root -p

Enter password: Mysql@1234

SHOW MASTER STATUS\G

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

File: mysql-bin.000022

Position: 155

Binlog\_Do\_DB:

Binlog\_Ignore\_DB:

Executed\_Gtid\_Set:

1 row in set (0.01 sec)

Step 2

\*\*\*\*\*\*

-create a new replication user in master server who can login master server from slave server.

[root@linux2 ~]# CREATE USER 'replica'@'192.168.1.61' IDENTIFIED WITH mysql\_native\_password BY 'Password@123';

[root@linux2 ~]# GRANT REPLICATION SLAVE ON \*.\* TO 'replica'@'192.168.1.61';

Select host, user from mysql.user;

it shows

------------------------------------

host | user |

------------------------------------

% | mysqlbackup51 |

192.168.1.61 | replilca |

localhost | mysql.infoschema |

......... |

----------------------------------------

try to connect to master using the newly created user(replica)

[root@linux3 ~]# mysql -u replica -h 192.168.1.51 -p

Enterpassword: ENter the password that is alloted to while creating the replica user which is mentioned after identified by

it shows

mysql>

it means u have logged in to master from slave using the replica credentials

Step 2 : [root@linux3 ~]#CONFIGURE THE SLAVE SERVER

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Like master server perform below changes on slave server.

Set the MySQL server to listen on the private IP

Set a unique server ID

Enable the binary logging

slave:/etc/my.cnf

cp /etc/my.cnf /etc/my.cnf\_orig

vi /etc/my.cnf

bind-address = 192.168.1.61

server-id = 2

log\_bin = mysql-bin # The value here is /mysql/logs/mysql-bin

# service mysqld restart

Step 3: Configure the parameters for slave server to connect to the master server.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

in slave terminal

mysql -uroot -p

Mysql@1234

mysql> STOP SLAVE;

mysql> CHANGE MASTER TO

->MASTER\_HOST='192.168.1.51',

->MASTER\_USER='replica',

->MASTER\_PASSWORD='Password@123',

->MASTER\_LOG\_FILE='mysql-bin.000023',

->MASTER\_LOG\_POS=155;

start slave;

show slave status\G;

show variables like '%uuid%';

testing

\*\*\*\*\*\*\*

in master server

Create database testreplica;

use testreplica;

CREATE TABLE `departments` (

`dept\_no` char(4) NOT NULL,

`dept\_name` varchar(40) NOT NULL,

PRIMARY KEY (`dept\_no`),

UNIQUE KEY `dept\_name` (`dept\_name`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;

got to slave dbs

show databases;

use testreplica;

show tables;

u can see departments table is created

so the master slave configuration is ready

Video2: How To Configure MySQL Master-Slave Replication on Ubuntu 18.04

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

lsb\_release -cd

getconf LONG\_BIT

hostname -I

gedit /etc/mysql/mysql.conf.d/mysql.cnf &>/dev/null

Video3:How to Configure MySQL Master-Slave Replication on Ubuntu Linux

88,109 viewsJan 6, 2015

vi /etc/mysql/my.cnf

make changes to srever id, log path, bind address

service mysql restart

in master create a replication user

mysql> create user 'replica'@'%' identified by 'password'

mysql>grant r,eplication slave on \*.\* to 'replica'@'%';

mysql>craete databse pets;

mysql>create table pets.cats (name varchar(20));

mysql>insert into pets.cats values ('fluffy');

mysql>selct \* from pets.cats;

exit

lets crqate a mysql data sanspshot using mysqldump

[vagrant01]mysqldump -u root -p --all-databses --master-data > masterdump.sql

using the --master-data option will cause the mysqldump to ride out the binary log position info in the dumpp file

the back up will have a changed master statement init

id we donrt use the --master-data, then u will have to manually put the master database intno read only mode, perform the dump and note the binary log file position by using show master stauts command

[vagrant01]grep CHANGE \*sql | head -1

it shows CHANGE MASTER TO MASTER LOG FILE='mysql-bin.000001', MASTER\_LOG\_POSITION=1177;

so u cansee it recoreded the binary log file and bianry log file position

lets copy that file to db02

[vagrant01]scp masterdump.sql 10.11.12.102

Now got slave or db02

[vagrant02]sudo apt-get install mysql-server

sudo /etc/mysql/my.cnf

edit bind adddress, server id (log\_path he did not mention)

sudo service mysql restart

mysql -u root -p

mysql>CHANGE MASTER TO

MASTER\_HOST='10.11.12.101'

MATER\_USER='replica'

MASTER\_PASSWORD='slavepassword'

mysql>exit

[vagrant02] mysql -u root < masterdump.sql

mysql - u root -

mysql>start slave

show slave status\G;

it shows master log file, position etc

Now go to master insert data come back to slave and u will find that the data is replicated in slave server