# MySQL 5.7.10多源复制

# 1.安装MySQL5.7.10

SHELL>wget <http://dev.mysql.com/get/Downloads/MySQL-5.7/mysql-community-server-5.7.10-1.el6.x86_64.rpm>

SHELL>wget <http://dev.mysql.com/get/Downloads/MySQL-5.7/mysql-community-client-5.7.10-1.el6.x86_64.rpm>

SHELL>wget <http://dev.mysql.com/get/Downloads/MySQL-5.7/mysql-community-common-5.7.10-1.el6.x86_64.rpm>

SHELL>wget <http://dev.mysql.com/get/Downloads/MySQL-5.7/mysql-community-libs-5.7.10-1.el6.x86_64.rpm>

SHELL>rpm -ivh [mysql-community-libs-5.7.10-1.el6.x86\_64.rpm](http://dev.mysql.com/get/Downloads/MySQL-5.7/mysql-community-libs-5.7.10-1.el6.x86_64.rpm) [mysql-community-common-5.7.10-1.el6.x86\_64.rpm](http://dev.mysql.com/get/Downloads/MySQL-5.7/mysql-community-common-5.7.10-1.el6.x86_64.rpm) [mysql-community-client-5.7.10-1.el6.x86\_64.rpm](http://dev.mysql.com/get/Downloads/MySQL-5.7/mysql-community-client-5.7.10-1.el6.x86_64.rpm) [mysql-community-server-5.7.10-1.el6.x86\_64.rpm](http://dev.mysql.com/get/Downloads/MySQL-5.7/mysql-community-server-5.7.10-1.el6.x86_64.rpm)

# 2.多源复制配置文件参数添加

在/etc/my.cnf  [mysqld]中添加两行

master\_info\_repository=TABLE  
relay\_log\_info\_repository=TABLE

注:my.cnf的配置这里就不完全贴出了。根据实际情况自行配置

# 3.初始化数据库

初始化数据库可以采用主库的二进制备份文件。也可以采用逻辑备份文件。这里建议采用二进制备份文件。

注:MySQL5.7版本开始MySQL数据库初始化已经可以在没有初始数据的时候直接启动。如果没有检测到数据文件时mysql会自动初始化数据。但是初始化之后会自动生成一个密码。可以在mysql的错误日志中找到。

mysql的初始化数据整理完成。my.cnf配置文件准备妥当，然后启动mysql服务

SHELL>service mysqld start

注:MySQL5.7  RPM安装完之后服务默认为mysqld 。可根据习惯改名 （/etc/init.d/mysqld）

# 4.升级mysql数据

因为mysql安装的5.7版本。但是恢复的备份数据为5.6版本。故此需要升级

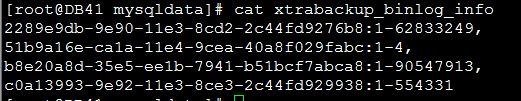
SHELL>mysql\_upgrade

SHELL>service mysqld restart

这一步骤很重要。单单的mysql\_upgrade不能完成mysql的升级。重启之后才能全部升级完毕

# 5.配置主从同步master1

.



SQL>set global gtid\_purged="2289e9db-9e90-11e3-8cd2-2c44fd9276b8:1-62833249,51b9a16e-ca1a-11e4-9cea-40a8f029fabc:1-4,b8e20a8d-35e5-ee1b-7941-b51bcf7abca8:1-90547913,c0a13993-9e92-11e3-8ce3-2c44fd929938:1-554331";

SQL>change master to master\_host="192.168.1.1",master\_user="rep",master\_password="reppass",master\_auto\_position=1 for channel "master1";

SQL>start slave for channel "master1";

SQL>show slave status for channel "master1“;

# 6.配置主从同步master2–N

除了第一个主库可以采用二进制备份初始化数据之外别的数据库只能用逻辑备份进行初始化。

注：在源库上逻辑备份的时候记得加参数--master-data=1  example:  mysqldump --master-data=1 --single-transaction --triggers  -B db1 db2 >master2dumpfile.sql

SHELL>grep -A 5 GTID\_PURGED master2dumpfile.sql

SHELL>grep -A 5 GTID\_PURGED master3dumpfile.sql

注：如果源库不是集群的话可以不用参数-A

SQL>source /data/master2dumpfile.sql;

SQL>source /data/master3dumpfile.sql;

SQL>stop slave for channel "master1";

SQL>show variables like "gtid\_purged%";

SQL>reset master;

SQL>set global gtid\_purged="2289e9db-9e90-11e3-8cd2-2c44fd9276b8:1-62833249,51b9a16e-ca1a-11e4-9cea-40a8f029fabc:1-4,b8e20a8d-35e5-ee1b-7941-b51bcf7abca8:1-90547913,c0a13993-9e92-11e3-8ce3-2c44fd929938:1-554331,03870568-18b4-ee1b-6085-0de4404fca4b:1-2847875,2991fd31-57b5-11e5-8974-a0d3c102a340:1-29,44594f04-a2f1-ee1a-722f-904f81f24eaa:1-957519";

注:这里设置的GTID是show variables like "gtid\_purged%"出来的Value和GREP出来的GTID以  ,  连接而成

SQL>change master to master\_host="192.168.1.1",master\_user="rep",master\_password="reppass",master\_auto\_position=1 for channel "master1";

SQL>change master to master\_host="192.168.1.2",master\_user="rep",master\_password="reppass",master\_auto\_position=1 for channel "master2";

SQL>change master to master\_host="192.168.1.3",master\_user="rep",master\_password="reppass",master\_auto\_position=1 for channel "master3";

SQL>start slave for channel "master1";

SQL>start slave for channel "master2";

SQL>start slave for channel "master3";

SQL>show slave status for channel "master1“;

注：show slave status for channel "master1“;可以查看单一信道的复制的详细状态

SQL>select \* from performance\_schema.replication\_applier\_status\_by\_worker;

注:这条SQL可以展示所有的复制信道的复制状态概况