**在线建立或重做mysql主从复制架构方法（GTID模式）**

# 一、主库授权

mysql>grant replication slave on \*.\* to 'rep'@'%' identified by 'password';

## mysqldump方式

### 第一步,数据的导出与导入和初始化从库

（以下操作，先在主库操作，后在从库操作因为mysql自带,不需要再做些什么,比较方便易用,不过需要强调一下,数据量太大的话,mysqldump就略显不足了,因为是逻辑备份和导入，导出导入时间耗费巨大,各位自己衡量.

首先，我们先从主库备份个整库备份：

|  |
| --- |
| /usr/local/mysql/bin/mysqldump --all-databases --single-transaction --master-data=2 --set-gtid-purged=on -F --triggers --routines --events --user=root -p --socket=/tmp/mysql.sock > all.sql |

备份做完了,那就是要拷到从库恢复了。

#使用GTID模式(由于主库备份时设置了--set-gtid-purged=on)的实例的导入,需要目标实例的GTID记录是空的,不然会报错,就类似这样:[root@pingtest1 ttt]# mysql -uroot -pqwe123 < all.sql

Warning: Using a password on the command line interface can be insecure.

ERROR 1840 (HY000) at line 24: @@GLOBAL.GTID\_PURGED can only be set when @@GLOBAL.GTID\_EXECUTED is empty.#提示GTID\_EXECUTED为非空,不能导入.#所以导入前要先重置下binlog

以下都是在从库操作：

mysql>reset master;#重置了binlog那就没问题了,重新执行

mysql -uroot -pqwe123 < all.sql#好了,导入完成后,我们检查一下,gtid模式要看看这个

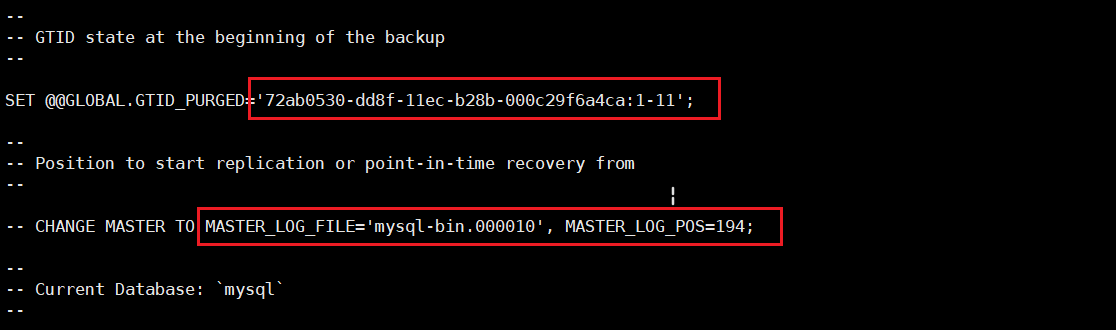
mysql>show variables like "%gtid%";

gtid\_purged | 16fdabc7-30f9-11e6-9234-0800273e5680:1-3216

### 第二步,配置主从同步（以下操作，都是在从库操作）

来看看all.sql的binlog位置,就文件的前50行,要是你觉得文件太大很难打开,more或者head -50都可以,就不要vim了,只看关键字

|  |
| --- |
| -- GTID state at the beginning of the backup  SET @@GLOBAL.GTID\_PURGED='2e0de4cf-be68-11e9-a353-000c2980d8b3:1-219';  -- Position to start replication or point-in-time recovery from  -- CHANGE MASTER TO MASTER\_LOG\_FILE='mysql-bin.000005', MASTER\_LOG\_POS=194;  -- Current Database: `mysql`  CREATE DATABASE /\*!32312 IF NOT EXISTS\*/ `mysql` /\*!40100 DEFAULT CHARACTER SET utf8 \*/; |



看到了,当前binlog位置是mysql-bin.000005,Position位置是194,gtid位置是2e0de4cf-be68-11e9-a353-000c2980d8b3:1-219

GTID模式下,则还需要多做一步,就是执行那个语句,设置gtid编号,用来告诉主库,我这些语句执行过了(**如果有reset master，可跳过此步**)

mysql>SET @@GLOBAL.GTID\_PURGED='2e0de4cf-be68-11e9-a353-000c2980d8b3:1-219';

就是跳过之前已经复制过的事务,因为GTID记录的事务是从1开始的,只会增加或变更UUID,不会重置计数器,所以就要手动设置跳过之前执行过(已经在备份里的结果)的事务.然后change master to一下,

mysql> change master to master\_host='172.60.51.91', master\_user='rep',master\_password='qwe123',master\_port=12308,master\_auto\_position=1;

mysql> start slave；

mysql> show processlist;

|  |
| --- |
| Master\_Server\_Id: 1021  Master\_UUID: 2e0de4cf-be68-11e9-a353-000c2980d8b3  Master\_Info\_File: mysql.slave\_master\_info  SQL\_Delay: 0  SQL\_Remaining\_Delay: NULL  Slave\_SQL\_Running\_State: Slave has read all relay log; waiting for more updates  Master\_Retry\_Count: 86400  Master\_Bind:  Last\_IO\_Error\_Timestamp:  Last\_SQL\_Error\_Timestamp:  Master\_SSL\_Crl:  Master\_SSL\_Crlpath:  Retrieved\_Gtid\_Set: 2e0de4cf-be68-11e9-a353-000c2980d8b3:576-930  Executed\_Gtid\_Set: 2e0de4cf-be68-11e9-a353-000c2980d8b3:1-930  Auto\_Position: 1 |

### 总结：

1. 在主库授权

|  |
| --- |
| mysql>grant replication slave on \*.\* to 'rep'@'%' identified by 'password'; |

2、先在主库dump

|  |
| --- |
| /usr/local/mysql/bin/mysqldump --all-databases --single-transaction --master-data=2 --set-gtid-purged=on -F --triggers --routines --events --user=root -p --socket=/tmp/mysql.sock > all.sql |

3、在从库导入

|  |
| --- |
| mysql> reset master #重置了binlog那就没问题了,重新执行 |
| mysql -uroot -pqwe123 < all.sql #好了,导入完成后,我们检查一下,gtid模式要看看这个 |
| mysql>show variables like "%gtid%"; |

4、在从库配置主从

|  |
| --- |
| mysql> change master to master\_host='172.60.51.91', master\_user='rep',master\_password='qwe123',master\_port=12308,master\_auto\_position=1;  mysql> start slave； |

## ****xtrabackup方式****

某些库很大,几十G几百G什么的也不用很意外,如果还用mysqldump就不科学了,导出导入耗费时间巨大,这个时候就必须靠xtrabackup这种物理拷贝的备份还原工具,加快速度进行,另一方面来说,从备份原理得出的结果来看,mysqldump备份得到的结果是备份开始时间的数据结果,而xtrabackup备份得到的结果是备份结束时间的数据结果,是属于比较新的数据,对于操作频繁的库用xtrabackup来做也是优势明显.

1. 主库授权
2. 全库备份

|  |
| --- |
| innobackupex --defaults-file=/etc/my.cnf --user=root --password=qwe123 --socket=/tmp/mysql.sock /tmp/back |

在从库开始恢复备份

xtrabackup\_binlog\_info,里面就记录了binlog位置和GTID值

|  |
| --- |
| cat xtrabackup\_binlog\_info  mysql-bin.000016 70613172 2e0de4cf-be68-11e9-a353-000c2980d8b3:1-1644 |

#关闭mysql服务

|  |
| --- |
| service mysql stop |

#先删了旧的数据文件

|  |
| --- |
| rm -rf /usr/local/mysql/data/\* |

#然后先apply-log

|  |
| --- |
| innobackupex --defaults-file='/etc/my.cnf' --apply-log /tmp/2019-08-15\_09-42-08/ |

#然后开始恢复

|  |
| --- |
| innobackupex --defaults-file='/etc/my.cnf' --copy-back /tmp/2019-08-15\_09-42-08/ |

#恢复完成后看一下,做完收尾工作

|  |
| --- |
| ll /usr/local/mysql/data  chown -R mysql.mysql /usr/local/mysql/data |

#启动mysql服务

|  |
| --- |
| service mysql start |

#尝试登陆一下,ok就完成了

|  |
| --- |
| mysql -uroot -pqwe123 |

4、在从库配置主从

|  |
| --- |
| mysql> reset master; #重置了binlog那就没问题了,重新执行  mysql> reset slave all; #做之前，别忘记先清空一下主从结构的记录  mysql> change master to master\_host='172.60.51.91', master\_user='rep',master\_password='qwe123',master\_port=12308,master\_auto\_position=1;  mysql> start slave; |

参考url:<https://blog.51cto.com/arthur376/1792551>

show processlist;

show master status\G

show slave hosts;

show slave status\G