linux 下 mysql 实现双向同步

作者：upaths
原文链接：https://ld246.com/article/1558084879167
来源网站：链滴

MySQL 双向同步

也被称为 **主主同步** ，即两个 MySQL 服务都是 Master，其中任意一个服务又是另一个服务的 Slave。

准备

服务器

MySQL服务器	IP地址
masterA	192.168.1.201
masterB	192.168.1.202

具体操作

注意

操作过程中注意两边数据的一致！！！

masterA 配置

my.cnf

```
[mysqld]

# 服务器唯一标识

server-id=1

```

```
# 二进制日志文件名

log-bin=mysql-bin

# 需要备份的数据库，多个数据库用，分隔

binlog-do-db=piumnl

# 需要复制的数据库，多个数据库用，分隔

replicate-do-db=piumnl

# 中继日志文件名

relay_log=mysqld-relay-bin

# 互为主从需要加入这一行

log-slave-updates=ON

# 禁用符号链接，防止安全风险，可不加

symbolic-links=0

# 可不加

# resolve - [Warning] Slave SQL: If a crash
# happens this configuration does not guarantee that the relay log info will be
# consistent, Error_code: 0

master-info-repository=table

relay-log-info-repository=table

relay-log-recovery=1


# 可不加

# 禁用 dns 解析，会使授权时使用的域名无效

skip-host-cache

skip-name-resolve


sql_mode=NO_ENGINE_SUBSTITUTION,STRICT_TRANS_TABLES
```

masterB 配置

my.cnf

```
# 不再解释各个配置项
 [mysqld]
```

```
server-id=2
log-bin=mysql-bin

binlog-do-db=piumnl
replicate-do-db=piumnl
relay_log=mysql-relay-bin
log-slave-updates=ON
symbolic-links=0

# resolve - [Warning] Slave SQL: If a crash happens this configuration does not
# guarantee that the relay log info will be consistent, Error_code: 0

master-info-repository=table
relay-log-info-repository=table
relay-log-recovery=1

skip-host-cache
skip-name-resolve

sql_mode=NO_ENGINE_SUBSTITUTION,STRICT_TRANS_TABLES
```

创建同步用户

masterA & masterB 都要创建同步用户：

```
create user 'rep'@'%' identified by 'rep';    # 创建一个账户

grant replication slave on *.* to 'rep'@'%';  # 授予该账户对任意数据库任意表的主从同步权限
```

备注：

1. Linux 下 MySQL 对 root@% 关闭了 grant_priv 权限
2. 此处备份用户帐号和密码可不一致，此处为了简化操作使用一样的帐号和密码

重启服务器

重启服务器

开启同步

masterA

1. 查看 masterB 状态

```
show master status\G;
```

此处需要关注 File 和 Position 值

2. 开启同步

```
stop slave;
```

```
# master_log_file 就是第一步操作的 File 值
# master_log_pos 就是第一步操作的 Position 值

change master to master_host=<master_hostname>, master_user=<rep_username>, master password=<rep_password>, master_log_file='mysql-log.000003', master_log_pos=154;

start slave;
```

3. 查看结果

```
show slave status\G;
# 查看最重要的两项，两个都必须为 Yes ，有一个为 No 都要去查看错误日志文件，看看什么地方在问题
# Slave_IO_Running: Yes
# Slave_SQL_Running: Yes
```

masterB

1. 重复 masterA 的操作

测试

分别在 masterA 和 masterB 中插入数据，并查看另一台服务器是否及时出现预期的数据

注：

1. 使用 <database>.<table> 方式进行插入、更新和删除操作，将不会备份。
2. 如果操作弄乱了 master 与 slave 的日志，可使用如下操作进行重置。

```
reset master; # 重置 master 的配置，包括二进制日志

reset slave; # 重置 slave 的配置
```