**mysql主从同步**

**角色分为2种：**

**数据库服务器 做主master库：被客户端存储数据访问的库**

**数据库服务器 做从slave库：同步主库的数据到本机**

**作用：实现数据的自动备份**

**从库的sql线程执行本机中继日志里的sql命令，不会记录在binglog日志文件里。**

主从同步概述

mysql主从同步

对指定库的异地同步

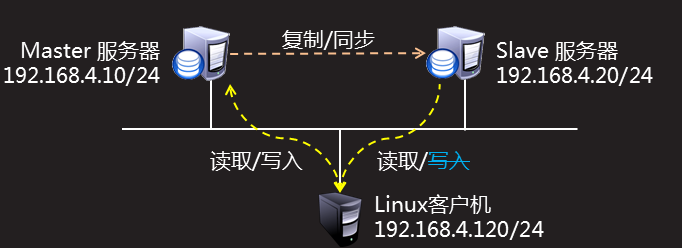
mysql主->从复制架构的实现

mysql服务器的只读权限

案例拓扑

一主、一从

单向复制时，建议将从库设为只读



主从同步原理

master，记录数据更改操作

启用binlog日志

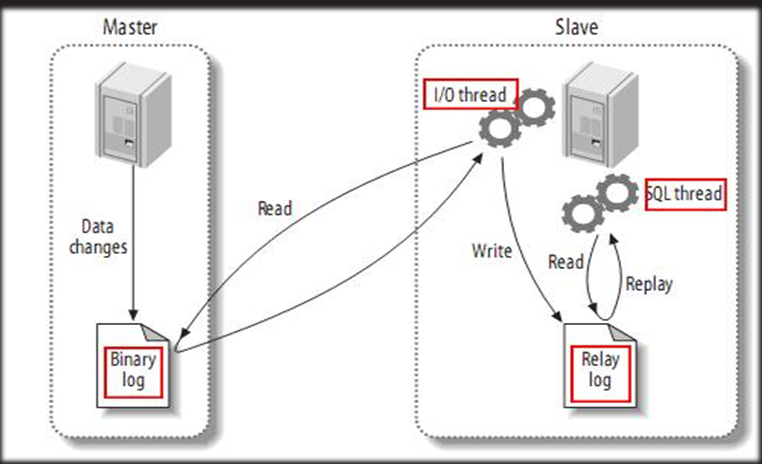
设置binlog日志格式

设置server\_id

slave运行2个线程

slave\_io：复制master主机 binlog日志文件里的sql到本机的relay-log文件里

slave\_sql：执行本机relay-log文件里的sql语句，重现master的数据操作



构建主从同步

基本构建思路

确保数据相同

从库必须要有主库上的数据

配置主服务器

启用binlog日志、授权用户、查看当前正使用的日志

配置从服务器

设置server\_id，指定主库信息

测试配置

客户端连接主库写入数据，在从库上也能查到

确保数据一致

master服务器

应包括希望同步的所有库

对采用myisam的库，可离线备份

mysqldump -uroot -p密码 -B 库名列表 > mytest.sql

slave服务器

离线导入由master提供的备份

清空同名库（若有的话）

scp dbsvr1:/root/mytest.sql ./直接scp远程拷贝

mysql -uroot -p < mytest.sql

配置主服务器

调整运行参数

启用binlog及允许同步（服务端）

vim /etc/my.cnf

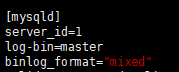
[mysqld]

log\_bin=日志名

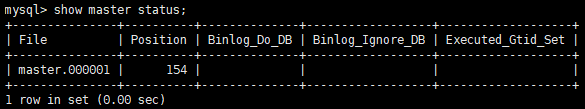
server\_id = id值

binlog\_format=”mixed”

systemctl restart mysqld







授权用户

允许replicater从192.168.4.0/24网段访问

对所有库（默认不允许对单个库）有同步权限

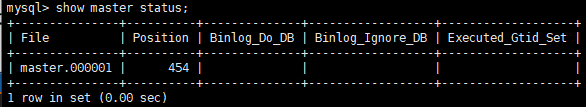
mysql>grant replication slave on \*.\* to 用户名@’从库ip地址’ identified by ‘密码’;



查看master状态

记录当前的日志文件名、偏移位置

mysql>show master status\G;



配置从服务器



调整运行参数

指定server\_id不允许与主库server\_id值相同

vim /etc/my.cnf

[mysqld]

server\_id=id值

systemctl restart mysqld





指定主库信息（首次是指定，第二次是修改）

mysql>change master to

->master\_host=’192.16.4.10’, //主库ip地址

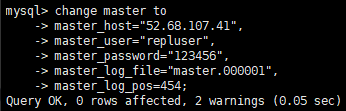
->master\_user=’replicater’, //主库授权用户名

->master\_password=’pwd123’, //授权用户密码

->master\_log\_file=’dbsvr1-bin.000004’, //日志文件

->master\_log\_pos=334;

mysql>start slave;//启动slave进程





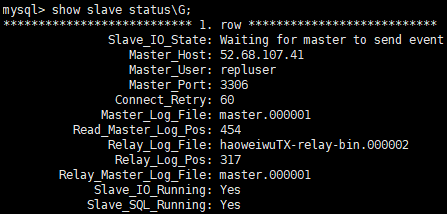
master信息会自动保存到/var/lib/mysql/master.info文件

以后要更改master信息时，应先stop slave;

查看slave状态

确认IO线程、sql线程都已运行

mysql>show slave status\G;



相关文件

|  |  |
| --- | --- |
| 文件名 | 说明 |
| master.info | 主库信息 |
| relay-log.info | 中继日志信息 |
| 主机名-relay-bin.xxxxxx | 中继日志 |
| 主机名-relay-bin.index | 索引文件 |

测试配置

在master上操纵数据

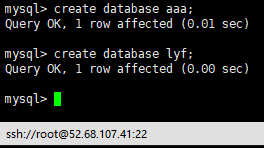
新建newdb库、newtbl表

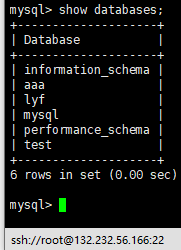
任意插入几条表记录

在slave上查看数据更改情况

确认新建的库、表

列出newtbl表的所有记录





让当前从库临时不同步主库上的内容

mysql>stop slave;

在从库修改主库信息

mysql>stop slave;

mysql>change master to 选项=”值”，选项=”值”;

mysql>start slave

常用配置选项

主库配置选项

适用与master 服务器（对所有的从库有影响）

|  |  |
| --- | --- |
| 选项 | 用途 |
| binlog\_do\_db=name | 设置master对哪些库记日志 |
| binlog\_ignore\_db=name | 设置master对哪些库不记日志 |

从库配置选项

适用于slave服务器（只对本机有影响）

|  |  |
| --- | --- |
| 选项 | 用途 |
| log\_slave\_updates | 记录从库更新，级联复制（A-B-C） |
| relay\_log=dbsvr2-relay-bin | 指定中继日志文件名 |
| replicate\_do\_db=mysql | 仅复制指定库，其他库将被忽略，此选项可设置多条（省略时复制所有库，多个库用逗号分隔） |
| replicate\_ignore\_db=test | 不复制哪些库，其他库将被忽略，ignore-db与do-db只选用其中一种 |

**主从同步模式**

结构类型

主从同步结构

基本应用

单向复制：主-->从

扩展应用

链式复制：主-->从-->从

互为主从：主<-->主

一主多从：从<--主-->从

配置主从从结构

拓扑结构

主从从



配置主库

1. 用户授权
2. 启用binlog日志
3. 重启服务

mysql>grant replication slave on \*.\* to 用户名@’从库IP地址’ identified by ‘密码’;

vim /etc/my.cnf

[mysqld]

log-bin=日志号

server\_id=id号

binlog\_format=’mixed’

systemctl restart mysqld

配置第1台从库

1. 修改配置文件
2. 用户授权
3. 指定主库信息
4. 启动slave从库进程

vim /etc/my.cnf

[mysqld]

server\_id=id号

log-bin=日志名

binlog\_format=’mixed’

log\_slave\_updates

systemctl restart mysqld

mysql>grant replication slave on \*.\* to 用户名@’第2台从库的IP地址’ identifief by ‘密码’;

mysql>change master to master\_host=’主库IP地址’,

->master\_user=’用户名’,

->master\_password=’密码’,

->master\_log\_file=’binlog日志文件名’

->master\_log\_pos=偏移量;

mysql>start slave;

mysql>show slave status\G;

配置第2台从库

1. 修改配置文件
2. 指定主库信息
3. 启动slave进程
4. 查看状态信息

vim /etc/my.cnf

server\_id=id号

systemctl restart mysqld

mysql>change master to master\_host=’第1台从库IP地址’,

->master\_user=’用户名’,

->master\_password=’密码’,

->master\_log\_file=’binlog日志文件名’

->master\_log\_pos=偏移量;

mysql>start slave;

mysql>show slave status\G;

测试配置

1. 在主库授权访问数据的用户
2. 访问主库，执行建库、建表、插入记录等操作
3. 访问第1台从库，可看到主库操作结果
4. 访问第2台从库，也可以看到主库操作结果

复制模式

复制模式介绍

异步复制（默认的复制模式）

主库执行完一次事务后，立即将结果返回给客户端，并不关心从库是否已经接受并处理

全同步复制

当主库执行完一次事务，且所有从库都执行了该事务后才返回给客户端

半同步复制

介于异步复制和全同步复制之间

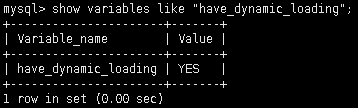
主库在执行完一次事务后，等待至少一个从库写到relay log中才返回给客户端

模式配置

查看是否允许动态加载模块

默认允许

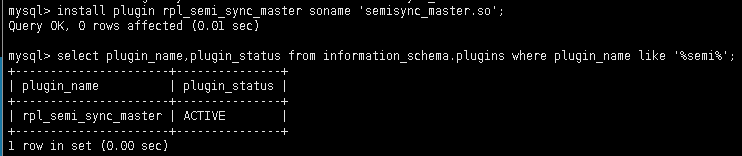
mysql>show variables like ‘have\_dynameic\_loading’;



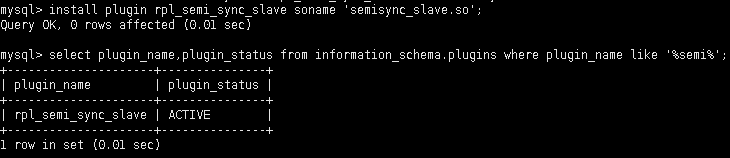
命令行加载插件

用户需要有super权限

mysql>install plugin rpl\_semi\_sync\_master soname ‘semisync\_master.so’; //主库上执行



mysql>install plugin rpl\_semi\_sync\_slave soname ‘semisync\_slave.so’; //从库上执行



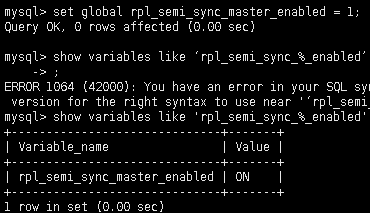
mysql>select plugin\_name,plugin\_status from information\_schema.plugins where plugin\_name like ‘%semi%’;



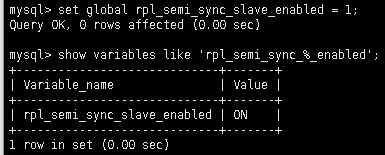
启用半同步模式

在安装完插件后，半同步复制默认是关闭的

mysql>set global rpl\_semi\_sync\_master\_enabled = 1; //在主库上执行



mysql>set global rpl\_semi\_sync\_slave\_enabled = 1; //在从库上执行



mysql>show variables like ‘rpl\_semi\_sync\_%\_enabled’; //查看

永久启用半同步复制

需要添加到主配置文件/etc/my.cnf

添加相关设置到[mysqld]部分

plugin-load=rpl\_semin\_sync\_master=semisync\_master.so

rpl\_semi\_sync\_master\_enabled=1 //主库配置



plugin-load=rpl\_semin\_sync\_slave=semisync\_slave.so

rpl\_semi\_sync\_slave\_enabled=1 //从库配置



在高可用架构下，master和slave需同时启动

以便在切换后能继续使用半同步复制

plugin-load=”rpl\_semi\_sync\_master=semisync\_master.so;rpl\_semi\_sync\_slave=semisync\_slave.so”

rpl\_semi\_sync\_master\_enabled=1

rpl\_semi\_sync\_slave\_enabled=1

