参考：<http://blog.csdn.net/mijinghjb/article/details/46605573>

主从配置步骤：  
 假设两台机器分别为192.168.0.202， 192.168.0.203，其中202做主，203做从。

1. 在两台[**MySQL**](http://lib.csdn.net/base/mysql)上创建同步用户。
   1. 202执行：  
      GRANT ALL PRIVILEGES ON \*.\* TO 'repl'@'192.168.0.203' IDENTIFIED BY 'repl' WITH GRANT OPTION;
   2. 203执行：

GRANT ALL PRIVILEGES ON \*.\* TO 'repl'@'192.168.0.202' IDENTIFIED BY 'repl' WITH GRANT OPTION;

让两个[**mysql**](http://lib.csdn.net/base/mysql)可以互相通过repl/repl访问对方。

1. 对两台机器，打开/etc/my.cnf mysql配置文件，在[mysqld]下打开以下选项：

log-bin=mysql-bin

innodb\_flush\_log\_at\_trx\_commit = 1  
sync\_binlog=1

binlog-do-db=bookmark  
binlog-do-db=debate  
binlog-ignore-db=test  
binlog-ignore-db=mysql

replicate-do-db=bookmark  
replicate-do-db=debate

其中binlog-do-db和replicate-do-db标记出需要进行同步的[**数据库**](http://lib.csdn.net/base/mysql)，对于多个数据库，重复选项多次即可，如上面的配置则表示同步bookmark和debate。binlog-ignore-db表示不进行同步的数据库，上面的配置表示不同步test, mysql库。

对于主mysql(202)，打开选项：  
server-id=1

对于203，从mysql，打开选项  
server-id=2

注意server-id选项必须打开，并且集群中每个mysql都要有一个单独的值，不能重复。一般设置主为1，从为2, 3, 4....。手册上说只要不相同什么整数都没有关系，可以设置为IP的值，如：202,

203.

3. 重新启动两个mysql. 分别输入：show master status; 命令，此时应该显示：

+------------------+----------+-----------------+------------------+

| File             | Position | Binlog\_Do\_DB    | Binlog\_Ignore\_DB |

+------------------+----------+-----------------+------------------+

| mysql-bin.000006 |       98 | bookmark,debate | test,mysql       |

+------------------+----------+-----------------+------------------+

1 row in set (0.00 sec)

输入：show slave status; 命令，应该显示：  
Empty set (0.00 sec)

表示还没有从。

1. 为了避免风险，主mysql现在最好不要对外提供服务，就是说主mysql的已有数据不要在动了。

主mysql(202), 执行下面的命令:  
FLUSH TABLES WITH READ LOCK;

锁定表，这个时候所有的commit都会失败。

执行：show master status;  
把输出结果记录下来，尤其是那个File和Position的值，后面要用。这里就是：mysql-bin.000006和98

把2个mysql都停掉，命令：./mysqladmin -u root shutdown

现在把master mysql(202)的数据复制到203上（me：手动拷贝），让slave的数据和master的同步。例如用sftp把202 data目录的bookmark, debate复制到203的data目录内。两边一定要保持一致。

5. 启动slave mysql(203)，执行命令：

CHANGE MASTER to MASTER\_HOST='192.168.0.202', MASTER\_PORT=3306, MASTER\_USER='repl', MASTER\_PASSWORD='repl', MASTER\_LOG\_FILE='mysql-bin.000006', MASTER\_LOG\_POS=98

其中master\_host就是主mysql, master\_log\_file, master\_log\_pos为上一步在master端执行show master status; 获得的File和Position值，此处换成那些值。注意数字是没有单引号引起来的。如果上面一步File和Position没有值，master\_log\_file='', master\_log\_pos=4

然后执行: start slave;

6. 启动master mysql(202).

在master端执行：unlock tables; 解除第4步的锁表。

在slave端执行：show slave status \G ;

察看Slave\_IO\_State，一般应该为"Waiting for master to send event"；此时Slave\_IO\_Running和Slave\_SQL\_Running应该为YES，

如果不是，在slave端执行: stop slave; start slave; 重启复制进程。Replicate\_Do\_DB为要复制的数据库，和my.cnf配置的相同。

在master端执行： show master status; 对比File, Position的值和上面slave端的Master\_Log\_File,Read\_Master\_Log\_Pos，两边相同则表示搭建完成。

7. [**测试**](http://lib.csdn.net/base/softwaretest)集群正确。

  1). 在master端同步的库里创建数据表abc，然后察看slave端是否同步创建了。  
  2). 在abc插入几条记录，察看slave端数据是否同步插入。  
  3). 更新abc的记录，察看slave端数据是否同步  
  4). 删除abc的记录，察看slave端数据是否同步  
  5). 停掉slave服务器，重新执行2,3,4的内容，重启slave，看slave端数据是否同步。  
   
  在执行上面的操作过程中，每执行一次都用第6步的方法察看master的File, Position值和slave的Master\_Log\_File,Read\_Master\_Log\_Pos是否相同，如果主从工作正常，这两个值在一次同步完成以后（slave status的Slave\_IO\_State显示：Waiting for master to send event）将会完全相同。

8. 日志检查

  如果有错误，检查mysql data目录的 ...machinname.err 文件。

9. 应用切换到slave

 如果master mysql坏掉了，需要将应用转移到slave段。首先在slave端执行：show master status; 把输出保存下来，然后再把应用切换过去。保存输出的File和Position值将可以用于以后从slave恢复master.

 具体从slave恢复master的办法：

 在master端执行：

CHANGE MASTER to MASTER\_HOST='192.168.0.203', MASTER\_PORT=3306, MASTER\_USER='repl', MASTER\_PASSWORD='repl', MASTER\_LOG\_FILE='mysql-bin.000006', MASTER\_LOG\_POS=98 ;

其中 MASTER\_HOST 为slave的ip, MASTER\_LOG\_FILE, MASTER\_LOG\_POS为切换到slave时记录下的File和Position值。

在master执行：start slave;

待master内容基本恢复以后（在master执行：show slave status \G; Slave\_IO\_State显示为Waiting for master to send event）

将应用切换回master。

待系统稳定后master执行：stop slave停止slave。如果要避免master重启以后自动开启slave，可以将master mysql的data文件夹的master.info文件删掉。

===================================================================================================================

补充，转载自51cto,nmshuishui：<http://nmshuishui.blog.51cto.com/1850554/1395587>，转载时请务必以超链接形式标明文章 [原始出处](http://nmshuishui.blog.51cto.com/1850554/1395587) 、作者信息

一、主从同步的作用

   1、数据分布

   2、负载均衡（load balancing)

   4、高可用性（high availability)和容错

二、主从同步原理

   mysql（MariaDB）的主从复制其实很简单，只要搞懂了原理，配置一个主从复制就是三下五除二的工夫

1、二进制日志

   想要做主从复制必须得有这个二进制日志，那二进制日志记录的是什么呢？二进制日志中保存的只是跟服务器修改相关的操作，也叫复制日志

   二进制日志为了避免日志文件过大，要进行不停地滚动，其主要有两个功能：

   （1）即时点恢复

   （2）复制

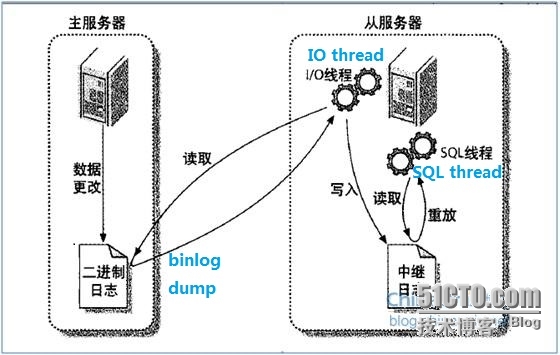
   因为二进制日志有即时点恢复功能，所以万一某天一个不小心失误把数据库删了，只要二进制日志文件保存完好，就可以把我们的数据恢复回来。可见二进制日志是非常重要的，因此二进制日志千万不要和你的数据库目录放到一起，而要另找一个位置保存

   今天我们要做的是利用他的第二个功能完成主从复制

2、中继日志

   中继日志和复制相关，中继日志是从服务器从主服务器上读取主服务器上二进制日志中的事件，每读一个过来，并保存到本地。

3、主从同步原理



   原理详解：

   1）主服务器发生数据更改操作，数据写入到数据文件和二进制文件中，然后去唤醒从服务器的I/O线程（IO thread）

   2）从服务器的IO thread来请求主服务器并把从主服务器的binlog dump线程返回结果写入到本地中继日志中

   3）从服务器的SQL thread每读一条中继日志，然后写一条到本地数据文件中

   4）此过程中，从服务器只是同步主服务器的数据，所以他无需开启二进制日志，而且从服务器最好设为只读模式

   5）主从复制并不是一个备份的过程，它只是一个数据同步的过程，因为，如果你在主服务器上删除一个库，那你会发现从服务器上也没有了

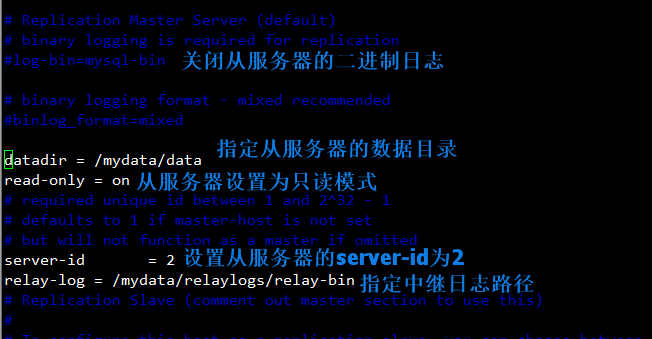
   6）从服务器并不能负载主服务器的写操作，它只可以负载主服务器的读操作；因为，不管你主服务器写多少，从服务器都会通过它的中继日志和主服务器的二进制日志同步到它本地的

三、主从同步的实现

1、修改主服务器的配置文件，我这里的数据目录在/mydata/data下，为了安全，我把二进制日志放到了/mydata/binlogs下，别忘了给这几个目录设置属主和属组为mysql用户，mysql组

[](http://s3.51cto.com/wyfs02/M00/24/2D/wKioL1NL5tPhDZQhAACGNoQylJA433.png)

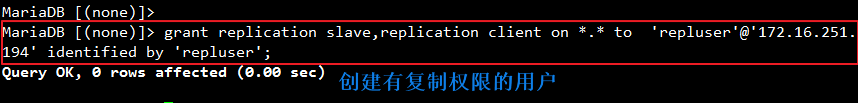
2、在从服务器上修改主配置文件，方法同上一步

[](http://s3.51cto.com/wyfs02/M01/24/2D/wKioL1NL523Q60FnAACpJeMS5uo355.png)

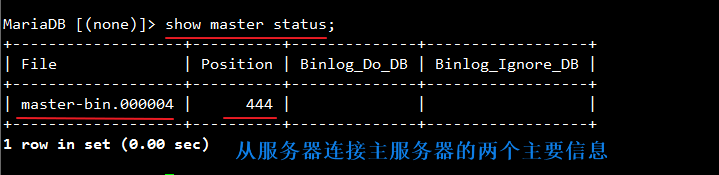
3、查看从服务器的中继日志是否启动成功

[](http://s3.51cto.com/wyfs02/M02/24/2C/wKiom1NL59qQrvfKAABSSR44tJo687.png)

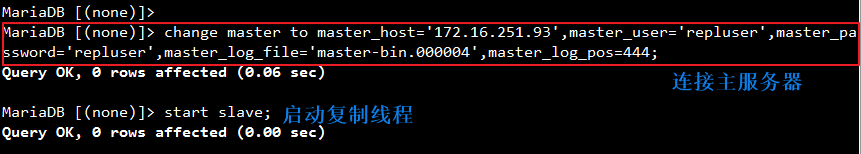
4、在主服务器上创建有复制权限的账号

[](http://s3.51cto.com/wyfs02/M02/24/2C/wKiom1NL6GWDCPnsAAAq6ksFitA533.png)

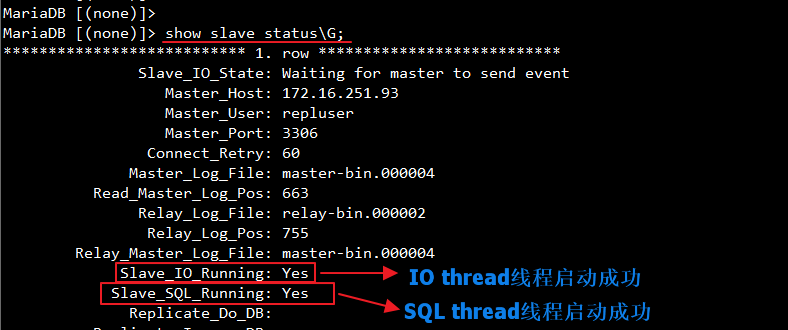
5、在主服务器上查看其状态信息

[](http://s3.51cto.com/wyfs02/M01/24/2D/wKioL1NL6GniGkKvAAAxheGBB2A943.png)

6、从服务器连接主服务器

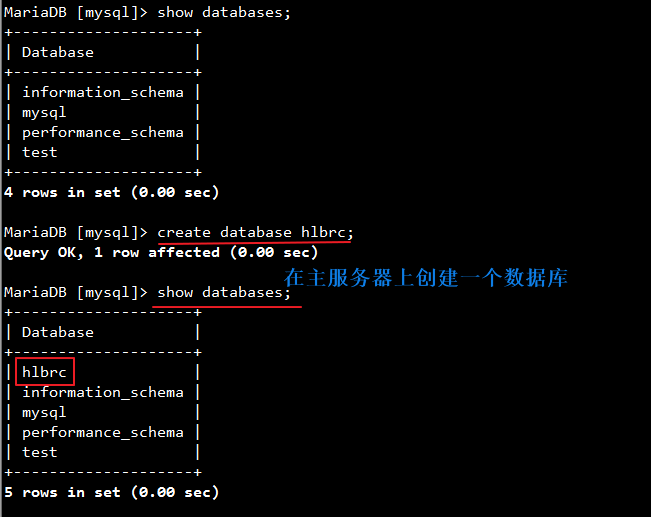
[](http://s3.51cto.com/wyfs02/M02/24/2D/wKioL1NL6KKjOnMMAAA9x71JKVg434.png)

7、查看从服务器的状态信息，检查是否连接主服务器成功

[](http://s3.51cto.com/wyfs02/M01/24/2C/wKiom1NL6RLjlJ-2AAAxheGBB2A374.png)

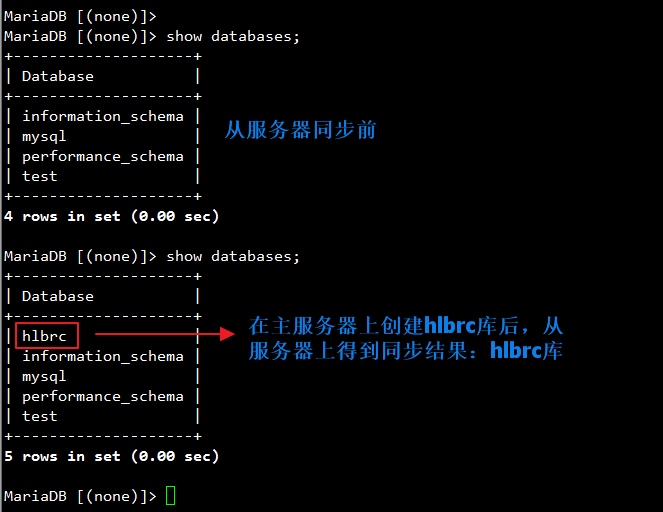
   这里的Slave\_IO\_Running和Slave\_SQL\_Running两个线程必须都为Yes才是正确的，出现了什么connecting啊还是no什么的都是没连接成功的。IO thread是去请求主服务器的二进制日志并写入到本地中继日志的，SQL thread是从中继日志写入到本地数据文件的。

8、在主服务器上创建一个数据库

[](http://s3.51cto.com/wyfs02/M01/24/2D/wKioL1NL6ejA7IShAABNlcLvfgA495.png)

9、在从服务器上启动复制线程，测试hlbrc这个库是否同步成功

|  |  |
| --- | --- |
| 1 | mysql>start slave； |

[](http://s3.51cto.com/wyfs02/M00/24/2D/wKioL1NL6kax7-PJAABzjio32Vs192.png)

四、主从同步配置过程总结

1、主服务器

   1）修改server-id

   2）启用二进制日志

   3）创建有复制权限的账号

2、从服务器

   1）修改server-id

   2）启用中继日志

   3）连接主服务器

   4）启程复制线程