[MySQL 高可用：主从配置或主主(双主)配置](http://blog.csdn.net/kk185800961/article/details/49235975)

2015-10-19 01:19 8360人阅读 [评论](http://blog.csdn.net/kk185800961/article/details/49235975#comments)(0) [收藏](javascript:void(0);) [举报](http://blog.csdn.net/kk185800961/article/details/49235975#report)

http://static.blog.csdn.net/images/category_icon.jpg 分类：

MYSQL（36） http://static.blog.csdn.net/images/arrow_triangle%20_down.jpg MYSQL 高可用（8） http://static.blog.csdn.net/images/arrow_triangle%20_down.jpg

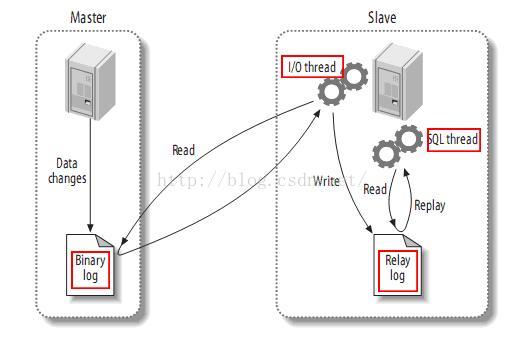
mysql 的主从复制可以实现mysql的多种高可用性，如数据库的读写分离 ，在线热备，负载均衡，数据分布 等。

mysql 主从同步原理：

1.  master 将操作记录到二进制日志(binary log)中；

2.  slave IO 线程 将master的binary log events读写到它的中继日志(relay log)；

3.  slave SQL进程读取中继日志，将重做记录数据到数据库中。



**下面配置：【主从复制】**

MySQL Version : 5.6.22  
主库：centos152 / 192.168.1.152  
从库：centos153 / 192.168.1.153

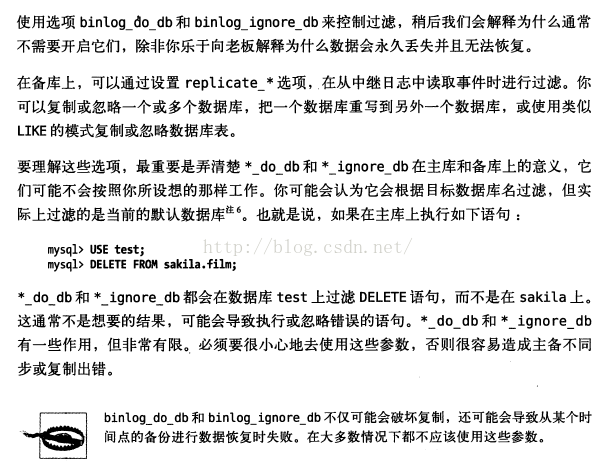
**配置 mysql 文件：**

【centos152 主库】  
[root@centos152 ~]# vi /etc/my.cnf  
[mysqld]  
basedir =/usr/local/mysql  
datadir =/usr/local/mysql/data  
port = 3306  
server\_id = 1  
log-bin= mysql-bin  
binlog\_format = mixed  
  
read-only=0  
#binlog-do-db=test  
binlog-ignore-db=mysql  
binlog-ignore-db=information\_schema  
binlog-ignore-db=performance\_schema  
auto-increment-offset=1  
auto-increment-increment=2  
  
#添加后重启mysql服务  
[root@centos152 ~]# service mysqld restart  
  
  
【centos153 从库】  
[root@centos153 ~]# vi /etc/my.cnf  
[mysqld]  
basedir =/usr/local/mysql  
datadir =/usr/local/mysql/data  
port = 3306  
server\_id = 2  
log-bin= mysql-bin  
binlog\_format = mixed  
  
read-only=0  
#replicate-do-db=test  
replicate-ignore-db=mysql  
replicate-ignore-db=information\_schema  
replicate-ignore-db=performance\_schema  
relay\_log=mysql-relay-bin  
log-slave-updates=on  
#auto-increment-offset=2  
#auto-increment-increment=2  
  
  
#添加后重启mysql服务  
[root@centos153 ~]# service mysqld restart

**说明：**

log-bin ：需要启用二进制日志  
server\_id : 用于标识不同的数据库服务器  
  
binlog-do-db : 需要记录到二进制日志的数据库  
binlog-ignore-db : 忽略记录二进制日志的数据库  
auto-increment-offset :该服务器自增列的初始值。  
auto-increment-increment :该服务器自增列增量。  
  
replicate-do-db ：指定复制的数据库  
replicate-ignore-db ：不复制的数据库  
relay\_log ：从库的中继日志，主库日志写到中继日志，中继日志再重做到从库。  
log-slave-updates ：该从库是否写入二进制日志，如果需要成为多主则可启用。只读可以不需要。  
  
如果为多主的话注意设置 auto-increment-offset 和 auto-increment-increment  
如上面为双主的设置：  
服务器 152 自增列显示为：1,3,5,7,……（offset=1，increment=2）  
服务器 153 自增列显示为：2,4,6,8,……（offset=2，increment=2）

**注意：文字图片来自《高性能 mysql》：**



#在 [root@centos152 ~] 操作几笔数据后，数据有些变化。

**[plain]** [view plain](http://blog.csdn.net/kk185800961/article/details/49235975) [copy](http://blog.csdn.net/kk185800961/article/details/49235975)

1. mysql> flush logs;
2. mysql> show master status;
3. +------------------+----------+--------------+---------------------------------------------+-------------------+
4. | File             | Position | Binlog\_Do\_DB | Binlog\_Ignore\_DB                            | Executed\_Gtid\_Set |
5. +------------------+----------+--------------+---------------------------------------------+-------------------+
6. | mysql-bin.000006 |      120 |              | mysql,information\_schema,performance\_schema |                   |
7. +------------------+----------+--------------+---------------------------------------------+-------------------+

File :当前正在记录的二进制日志文件  
Position ：记录偏移量，日志 mysql-bin.000006 所记录到的位置。  
Binlog\_Do\_DB ：要记录日志的数据库  
Binlog\_Ignore\_DB ：不记录日志的数据库  
Executed\_Gtid\_Set ：已执行的事务ID

**二进制日志情况：**

**[plain]** [view plain](http://blog.csdn.net/kk185800961/article/details/49235975) [copy](http://blog.csdn.net/kk185800961/article/details/49235975)

1. mysql> show variables like '%log\_bin%';
2. +---------------------------------+---------------------------------------+
3. | Variable\_name                   | Value                                 |
4. +---------------------------------+---------------------------------------+
5. | log\_bin                         | ON                                    |
6. | log\_bin\_basename                | /usr/local/mysql/data/mysql-bin       |
7. | log\_bin\_index                   | /usr/local/mysql/data/mysql-bin.index |
8. | log\_bin\_trust\_function\_creators | OFF                                   |
9. | log\_bin\_use\_v1\_row\_events       | OFF                                   |
10. | sql\_log\_bin                     | ON                                    |
11. +---------------------------------+---------------------------------------+

**日志位置：**

**[plain]** [view plain](http://blog.csdn.net/kk185800961/article/details/49235975) [copy](http://blog.csdn.net/kk185800961/article/details/49235975)

1. [root@centos152 ~]# ll /usr/local/mysql/data/mysql-bin\*
2. -rw-rw---- 1 mysql mysql 701 10-15 20:30 /usr/local/mysql/data/mysql-bin.000001
3. -rw-rw---- 1 mysql mysql 167 10-15 20:58 /usr/local/mysql/data/mysql-bin.000002
4. -rw-rw---- 1 mysql mysql 167 10-15 21:02 /usr/local/mysql/data/mysql-bin.000003
5. -rw-rw---- 1 mysql mysql 167 10-15 21:02 /usr/local/mysql/data/mysql-bin.000004
6. -rw-rw---- 1 mysql mysql 581 10-18 22:42 /usr/local/mysql/data/mysql-bin.000005
7. -rw-rw---- 1 mysql mysql 120 10-18 22:43 /usr/local/mysql/data/mysql-bin.000006
8. -rw-rw---- 1 mysql mysql 114 10-18 22:43 /usr/local/mysql/data/mysql-bin.index

#现在禁止操作主数据库！防止日志有变化，保证主从数据初始状态一致！

**[plain]** [view plain](http://blog.csdn.net/kk185800961/article/details/49235975) [copy](http://blog.csdn.net/kk185800961/article/details/49235975)

1. mysql> flush tables with read lock;

**备份主库**(执行以下一行脚本即可)

**[plain]** [view plain](http://blog.csdn.net/kk185800961/article/details/49235975) [copy](http://blog.csdn.net/kk185800961/article/details/49235975)

1. [root@centos152 ~]# mysqldump -u root -p test -F > /tmp/test\_backup\_`date +%Y\_%m\_%d\_%H\_%M\_%S`.sql
2. [root@centos152 ~]# mysqldump -u root -p --databases test --lock-all-tables --flush-logs> /tmp/test\_backup\_`date +%Y\_%m\_%d\_%H\_%M\_%S`.sql

**查看当前日志记录的位置：**

**[plain]** [view plain](http://blog.csdn.net/kk185800961/article/details/49235975) [copy](http://blog.csdn.net/kk185800961/article/details/49235975)

1. mysql> show master status \G;
2. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
3. File: mysql-bin.000009
4. Position: 120
5. Binlog\_Do\_DB:
6. Binlog\_Ignore\_DB: mysql,information\_schema,performance\_schema
7. Executed\_Gtid\_Set:
8. 1 row in set (0.00 sec)

**备份完后，可以解锁了**

**[plain]** [view plain](http://blog.csdn.net/kk185800961/article/details/49235975) [copy](http://blog.csdn.net/kk185800961/article/details/49235975)

1. mysql> unlock tables;

**在主服务器拷贝备份到从服务器：**

**[plain]** [view plain](http://blog.csdn.net/kk185800961/article/details/49235975) [copy](http://blog.csdn.net/kk185800961/article/details/49235975)

1. [root@centos152 ~]# scp -r /tmp/test\_backup\_2015\_10\_18\_23\_33\_30.sql root@192.168.1.153:/tmp
3. #不行则先关闭主从服务器防火墙
4. /etc/init.d/iptables status
5. /etc/init.d/iptables stop

**在从库中还原数据库：**

**[plain]** [view plain](http://blog.csdn.net/kk185800961/article/details/49235975) [copy](http://blog.csdn.net/kk185800961/article/details/49235975)

1. [root@centos153 ~]# mysql -u root -p test< /tmp/test\_backup\_2015\_10\_18\_23\_33\_30.sql

**主库创建用于连接到本地的用户：**

**[plain]** [view plain](http://blog.csdn.net/kk185800961/article/details/49235975) [copy](http://blog.csdn.net/kk185800961/article/details/49235975)

1. [root@centos152 ~]# grant replication slave on \*.\* to 'repl\_user'@'192.168.1.153' identified by 'slave@153';

**在从库中连接到主库：**

**[plain]** [view plain](http://blog.csdn.net/kk185800961/article/details/49235975) [copy](http://blog.csdn.net/kk185800961/article/details/49235975)

1. mysql>
2. CHANGE MASTER TO
3. MASTER\_HOST='192.168.1.152',
4. MASTER\_USER='repl\_user',
5. MASTER\_PASSWORD='slave@153',
6. MASTER\_LOG\_FILE='mysql-bin.000009',
7. MASTER\_LOG\_POS=120;

**重启从库mysql服务**  
[root@centos153 ~]# service mysqld restart

**查看从库同步信息：**

**[plain]** [view plain](http://blog.csdn.net/kk185800961/article/details/49235975) [copy](http://blog.csdn.net/kk185800961/article/details/49235975)

1. mysql> show slave status \G;
2. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
3. Slave\_IO\_State: Waiting for master to send event
4. Master\_Host: 192.168.1.152
5. Master\_User: repl\_user
6. Master\_Port: 3306
7. Connect\_Retry: 60
8. Master\_Log\_File: mysql-bin.000009
9. Read\_Master\_Log\_Pos: 120
10. Relay\_Log\_File: mysql-relay-bin.000003
11. Relay\_Log\_Pos: 283
12. Relay\_Master\_Log\_File: mysql-bin.000009
13. Slave\_IO\_Running: Yes
14. Slave\_SQL\_Running: Yes
15. Replicate\_Do\_DB:
16. Replicate\_Ignore\_DB: mysql,information\_schema,performance\_schema

可以看到， IO读写线程 Slave\_IO\_Running 和 SQL重做线程Slave\_SQL\_Running 都为 yes，表示正常执行！

**现在测试同步情况：**

**[plain]** [view plain](http://blog.csdn.net/kk185800961/article/details/49235975) [copy](http://blog.csdn.net/kk185800961/article/details/49235975)

1. #主库创建测试表
2. create table tabdemo(
3. id int primary key auto\_increment,
4. value int default 0
5. ) auto\_increment= 1 engine=innodb default charset=utf8;

8. insert into tabdemo(value) values(1),(1),(1),(1),(1);

11. select \* from tabdemo;
12. +----+-------+
13. | id | value |
14. +----+-------+
15. |  1 |     1 |
16. |  3 |     1 |
17. |  5 |     1 |
18. |  7 |     1 |
19. |  9 |     1 |
20. +----+-------+

主库操作，主库和从库都有数据了！id 编号与上面所说的一样！~现在主从复制已经配置成功！~

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

**下面配置：【主主复制】**

**更改配置文件：**

**【centos152 原主库】更改配置文件**  
[root@centos152 ~]# vi /etc/my.cnf  
[mysqld]  
basedir =/usr/local/mysql  
datadir =/usr/local/mysql/data  
port = 3306  
server\_id = 1  
log-bin= mysql-bin  
binlog\_format = mixed  
  
read-only=0  
#binlog-do-db=test  
binlog-ignore-db=mysql  
binlog-ignore-db=information\_schema  
binlog-ignore-db=performance\_schema  
auto-increment-offset=1  
auto-increment-increment=2  
  
#主主复制的从库设置（新增）  
#replicate-do-db=test  
replicate-ignore-db=mysql  
replicate-ignore-db=information\_schema  
replicate-ignore-db=performance\_schema  
relay\_log=mysql-relay-bin  
log-slave-updates=on  
  
  
**【centos153 原从库】更改配置文件**  
[root@centos153 ~]# vi /etc/my.cnf  
[mysqld]  
basedir =/usr/local/mysql  
datadir =/usr/local/mysql/data  
port = 3306  
server\_id = 2  
log-bin= mysql-bin  
binlog\_format = mixed  
  
read-only=0  
#replicate-do-db=test  
replicate-ignore-db=mysql  
replicate-ignore-db=information\_schema  
replicate-ignore-db=performance\_schema  
relay\_log=mysql-relay-bin  
log-slave-updates=on  
  
#主主复制的主库设置（新增）  
#binlog-do-db=test  
binlog-ignore-db=mysql  
binlog-ignore-db=information\_schema  
binlog-ignore-db=performance\_schema  
auto-increment-offset=2  
auto-increment-increment=2  
  
  
**添加后都重启mysql服务：**  
[root@centos153 ~]# service mysqld restart

**原从库(153)中创建用于连接的用户：**

**[plain]** [view plain](http://blog.csdn.net/kk185800961/article/details/49235975) [copy](http://blog.csdn.net/kk185800961/article/details/49235975)

1. [root@centos153 ~]# grant replication slave on \*.\* to 'repl\_user'@'192.168.1.152' identified by 'slave@152';

**原从库(153)查看当前日志记录的位置：**

**[plain]** [view plain](http://blog.csdn.net/kk185800961/article/details/49235975) [copy](http://blog.csdn.net/kk185800961/article/details/49235975)

1. mysql> show master status \G;
2. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
3. File: mysql-bin.000005
4. Position: 120
5. Binlog\_Do\_DB:
6. Binlog\_Ignore\_DB: mysql,information\_schema,performance\_schema
7. Executed\_Gtid\_Set:
8. 1 row in set (0.00 sec)

**在主库(152)中连接到原从库(153)，之前的从库现在也将是主库：**

**[plain]** [view plain](http://blog.csdn.net/kk185800961/article/details/49235975) [copy](http://blog.csdn.net/kk185800961/article/details/49235975)

1. mysql>
2. CHANGE MASTER TO
3. MASTER\_HOST='192.168.1.153',
4. MASTER\_USER='repl\_user',
5. MASTER\_PASSWORD='slave@152',
6. MASTER\_LOG\_FILE='mysql-bin.000005',
7. MASTER\_LOG\_POS=120;

**重启从库(152)mysql服务：**  
[root@centos152 ~]# service mysqld restart

**查看从库(152)的同步信息：**

**[plain]** [view plain](http://blog.csdn.net/kk185800961/article/details/49235975) [copy](http://blog.csdn.net/kk185800961/article/details/49235975)

1. mysql> show slave status \G;
2. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
3. Slave\_IO\_State: Waiting for master to send event
4. Master\_Host: 192.168.1.153
5. Master\_User: repl\_user
6. Master\_Port: 3306
7. Connect\_Retry: 60
8. Master\_Log\_File: mysql-bin.000005
9. Read\_Master\_Log\_Pos: 120
10. Relay\_Log\_File: mysql-relay-bin.000003
11. Relay\_Log\_Pos: 283
12. Relay\_Master\_Log\_File: mysql-bin.000005
13. Slave\_IO\_Running: Yes
14. Slave\_SQL\_Running: Yes
15. Replicate\_Do\_DB:
16. Replicate\_Ignore\_DB: mysql,information\_schema,performance\_schema

**现在已经配置好双主了！现在相互在两个数据库中都操作数据：**

**[plain]** [view plain](http://blog.csdn.net/kk185800961/article/details/49235975) [copy](http://blog.csdn.net/kk185800961/article/details/49235975)

1. #当前在原来的从库(153)操作数据。
3. insert into tabdemo(value) values(2),(2),(2),(2),(2);
5. #数据同步正常了！
6. select \* from tabdemo;
7. +----+-------+
8. | id | value |
9. +----+-------+
10. |  1 |     1 |
11. |  3 |     1 |
12. |  5 |     1 |
13. |  7 |     1 |
14. |  9 |     1 |
15. | 10 |     2 |
16. | 12 |     2 |
17. | 14 |     2 |
18. | 16 |     2 |
19. | 18 |     2 |
20. +----+-------+
22. 在原来的主库(152)操作数据。
23. insert into tabdemo(value) values(3),(3),(3),(3),(3);
25. #也正常！
26. select \* from tabdemo;
27. +----+-------+
28. | id | value |
29. +----+-------+
30. |  1 |     1 |
31. |  3 |     1 |
32. |  5 |     1 |
33. |  7 |     1 |
34. |  9 |     1 |
35. | 10 |     2 |
36. | 12 |     2 |
37. | 14 |     2 |
38. | 16 |     2 |
39. | 18 |     2 |
40. | 19 |     3 |
41. | 21 |     3 |
42. | 23 |     3 |
43. | 25 |     3 |
44. | 27 |     3 |
45. +----+-------+

至此，测试完毕！~注意参考：[学一点 mysql 双机异地热备份—-mysql主从，主主备份原理及实践](http://blog.csdn.net/kk185800961/article/details/49082499)

顶

1

踩