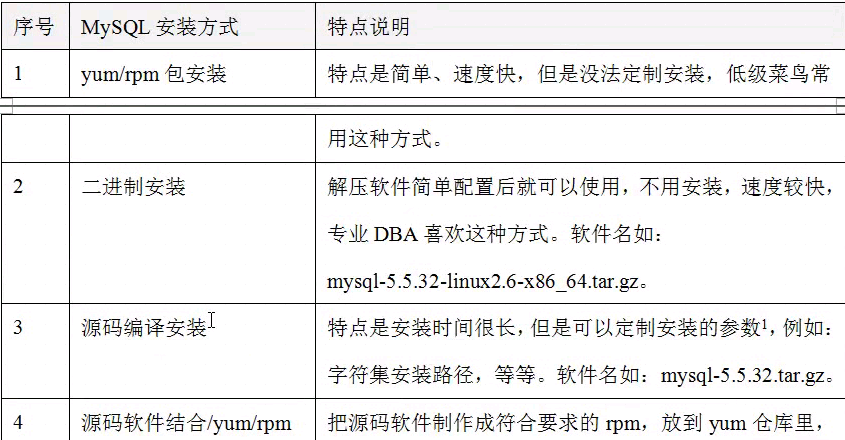
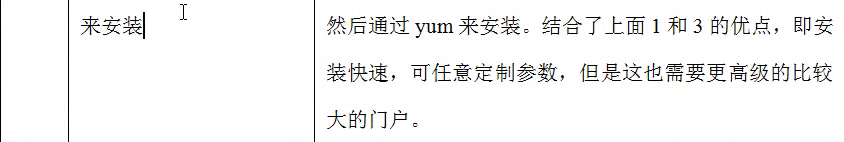
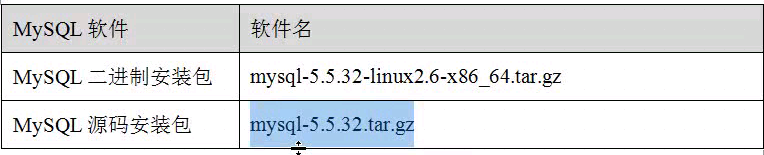
Mysql数据库应用实践

# mysql安装方式

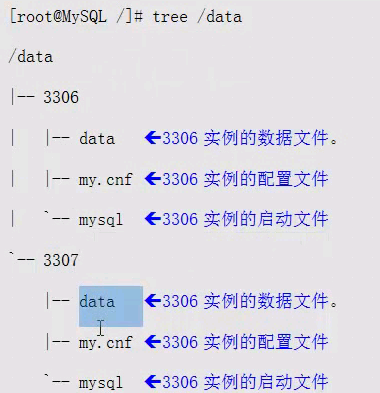




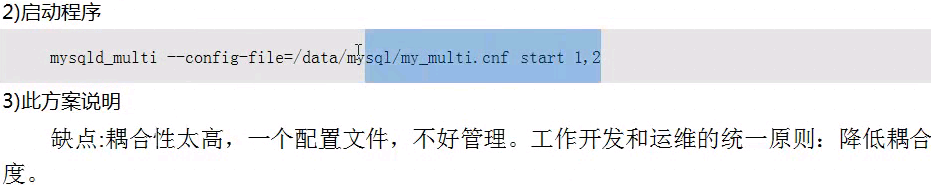


## 源码安装多实例

### 1.1配置方案一



### 1.2配置方案二



### 1.3源码安装

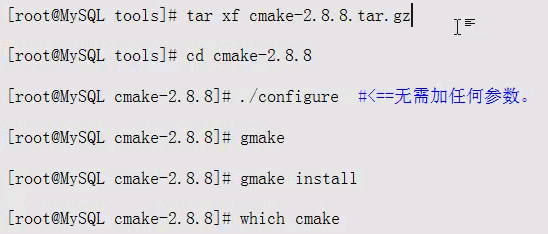
#### 安装依赖包

[root@mysql-db-server opt]# yum install ncurses-devel libaio-devel –y

[root@mysql-db-server cmake-2.8.12.2]# rpm -qa ncurses-devel libaio-devel

#### 安装编译mysql需要的软件

[root@mysql-db-server opt]# ll cmake-2.8.12.2.tar.gz



#### 创建mysql用户和组

[root@mysql-db-server cmake-2.8.12.2]# useradd mysql -s /sbin/nologin –M

#### 编译安装mysql

[root@mysql-db-server opt]# tar xf mysql-5.5.48.tar.gz

[root@mysql-db-server opt]# cd mysql-5.5.48

cmake \

-DCMAKE\_INSTALL\_PREFIX=/apps/mysql-5.5.48 \

-DMYSQL\_DATADIR=/apps/mysql-5.5.48/data \

-DSYSCONFDIR=/etc \

-DEXTRA\_CHARSETS=gbk,gb2312,utf8,ascii \

-DWITH\_MYISAM\_STORAGE\_ENGINE=1 \

-DWITH\_INNOBASE\_STORAGE\_ENGINE=1 \

-DWITH\_FEDERATED\_STORAGE\_ENGINE=1 \

-DWITH\_BLACKHOLE\_STORAGE\_ENGINE=1 \

-DWITHOUT\_EXAMPLE\_STORAGE\_ENGINE=1 \

-DWITH\_PARTITION\_STORAGE\_ENGINE=1 \

-DWITH\_READLINE=1 \

-DWITH\_FAST\_MUTEXES=1 \

-DWITH\_ZLIB=bundled \

-DWITH\_EMBEDDED\_SERVER=1 \

-DMYSQL\_UNIX\_ADDR=/apps/mysql-5.5.48/tmp/mysql/mysql.sock \

-DMYSQL\_TCP\_PORT=3306 \

-DENABLED\_LOCAL\_INFILE=ON \

-DDEFAULT\_CHARSET=utf8 \

-DDEFAULT\_COLLATION=utf8\_general\_ci \

-DWITH\_DEBUG=0

[root@mysql-db-server mysql-5.5.48]# make && make install

[root@mysql-db-server mysql-5.5.48]# ln -s /apps/mysql-5.5.48/ /apps/mysql

[root@mysql-db-server mysql-5.5.48]# ll support-files/my\*.cnf

-rw-r--r-- 1 root root 4743 2016-04-07 16:39 support-files/my-huge.cnf

-rw-r--r-- 1 root root 19807 2016-04-07 16:39 support-files/my-innodb-heavy-4G.cnf

-rw-r--r-- 1 root root 4717 2016-04-07 16:39 support-files/my-large.cnf

-rw-r--r-- 1 root root 4728 2016-04-07 16:39 support-files/my-medium.cnf

-rw-r--r-- 1 root root 2892 2016-04-07 16:39 support-files/my-small.cnf

[root@mysql-db-server data]# mkdir -p /data/{3306,3307}/data

#### 创建配置文件

**[root@mysql-db-server data]# vi /data/3307/data/my.cnf**

[client]

port = 3307

socket = /data/3307/mysql.sock

[mysql]

no-auto-rehash

[mysqld]

user = mysql

port = 3307

basedir = /apps/mysql

datadir=/data/3307/data

open\_files\_limit = 1024

back\_log = 600

max\_connections = 800

max\_connect\_errors = 3000

table\_cache = 614

external-locking = FALSE

max\_allowed\_packet = 8M

thread\_cache\_size = 100

thread\_concurrency = 2

query\_cache\_size = 4M

query\_cache\_limit = 2M

query\_cache\_min\_res\_unit = 2k

socket = /data/3307/mysql.sock

sort\_buffer\_size = 2M

join\_buffer\_size = 2M

default-storage-engine = innodb

lower\_case\_table\_names=1

thread\_stack = 192K

transaction\_isolation = REPEATABLE-READ

tmp\_table\_size = 64M

slow\_query\_log

long\_query\_time = 1

log-error = /data/3307/error.log

log-slow-queries = /data/3307/slow.log

server-id = 3

pid-file = /data/3307/mysql.pid

#log-bin = /data/3307/mysql-bin

relay-log = /data/3307/relay-bin

relay-log-info-file = /data/3307/relay-log.info

innodb\_file\_per\_table=1

binlog\_cache\_size = 2M

max\_heap\_table\_size = 64M

ft\_min\_word\_len = 4

binlog\_format=mixed

key\_buffer\_size = 8M

read\_buffer\_size = 2M

read\_rnd\_buffer\_size = 2M

bulk\_insert\_buffer\_size = 64M

myisam\_sort\_buffer\_size = 128M

myisam\_max\_sort\_file\_size = 10G

myisam\_repair\_threads = 1

myisam\_recover

innodb\_additional\_mem\_pool\_size = 16M

innodb\_buffer\_pool\_size = 200M

innodb\_data\_file\_path = ibdata1:10M:autoextend

innodb\_file\_io\_threads = 8

innodb\_thread\_concurrency = 16

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

innodb\_log\_buffer\_size = 16M

innodb\_log\_file\_size = 512M

innodb\_log\_files\_in\_group = 3

innodb\_max\_dirty\_pages\_pct = 60

innodb\_lock\_wait\_timeout = 120

[mysqldump]

quick

max\_allowed\_packet = 256M

[mysqld\_safe]

open-files-limit = 8192

pid-file = /data/3307/mysql.pid

log-error = /data/3307/mysql\_3307.err

**[root@mysql-db-server data]# vi /data/3306/data/my.cnf**

[mysql]

no-auto-rehash

[mysqld]

user = mysql

port = 3306

basedir = /apps/mysql

datadir=/data/3306/data

open\_files\_limit = 1024

back\_log = 600

max\_connections = 800

max\_connect\_errors = 3000

table\_cache = 614

external-locking = FALSE

max\_allowed\_packet = 8M

thread\_cache\_size = 100

thread\_concurrency = 2

query\_cache\_size = 4M

query\_cache\_limit = 2M

query\_cache\_min\_res\_unit = 2k

socket = /data/3306/mysql.sock

sort\_buffer\_size = 2M

join\_buffer\_size = 2M

default-storage-engine = innodb

lower\_case\_table\_names=1

thread\_stack = 192K

transaction\_isolation = REPEATABLE-READ

tmp\_table\_size = 64M

slow\_query\_log

long\_query\_time = 1

log-error = /data/3306/error.log

log-slow-queries = /data/3306/slow.log

server-id = 1

"my.cnf" 76L, 1792C 39,1 9%

[client]

port = 3306

socket = /data/3306/mysql.sock

[mysql]

no-auto-rehash

[mysqld]

user = mysql

port = 3306

basedir = /apps/mysql

datadir=/data/3306/data

open\_files\_limit = 1024

back\_log = 600

max\_connections = 800

max\_connect\_errors = 3000

table\_cache = 614

external-locking = FALSE

max\_allowed\_packet = 8M

thread\_cache\_size = 100

thread\_concurrency = 2

query\_cache\_size = 4M

query\_cache\_limit = 2M

query\_cache\_min\_res\_unit = 2k

socket = /data/3306/mysql.sock

sort\_buffer\_size = 2M

join\_buffer\_size = 2M

default-storage-engine = innodb

default\_table\_type = InnoDB

lower\_case\_table\_names=1

thread\_stack = 192K

transaction\_isolation = REPEATABLE-READ

tmp\_table\_size = 64M

slow\_query\_log

long\_query\_time = 1

log\_long\_format

log-error = /data/3306/error.log

log-slow-queries = /data/3306/slow.log

server-id = 1

pid-file = /data/3306/mysql.pid

#log-bin = /data/3306/mysql-bin

relay-log = /data/3306/relay-bin

relay-log-info-file = /data/3306/relay-log.info

innodb\_file\_per\_table=1

binlog\_cache\_size = 2M

max\_heap\_table\_size = 64M

ft\_min\_word\_len = 4

binlog\_format=mixed

key\_buffer\_size = 8M

read\_buffer\_size = 2M

read\_rnd\_buffer\_size = 2M

bulk\_insert\_buffer\_size = 64M

myisam\_sort\_buffer\_size = 128M

myisam\_max\_sort\_file\_size = 10G

myisam\_repair\_threads = 1

myisam\_recover

innodb\_additional\_mem\_pool\_size = 16M

innodb\_buffer\_pool\_size = 200M

innodb\_data\_file\_path = ibdata1:10M:autoextend

innodb\_file\_io\_threads = 8

innodb\_thread\_concurrency = 16

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

innodb\_log\_buffer\_size = 16M

innodb\_log\_file\_size = 512M

innodb\_log\_files\_in\_group = 3

innodb\_max\_dirty\_pages\_pct = 60

innodb\_lock\_wait\_timeout = 120

[mysqldump]

quick

max\_allowed\_packet = 256M

[mysqld\_safe]

open-files-limit = 8192

pid-file = /data/3306/mysql.pid

log-error = /data/3306/mysql\_3306.err

#### 调整启动文件

3306：

#!/bin/bash

mysql\_port=3306

mysql\_username="root"

mysql\_password="root"

function\_start\_mysql()

{

printf "Starting MySQL...\n"

/bin/sh /apps/mysql/bin/mysqld\_safe --defaults-file=/data/${mysql\_port}/my.cnf 2>&1 > /dev/null &

}

function\_stop\_mysql()

{

printf "Stoping MySQL...\n"

/apps/mysql/bin/mysqladmin -u ${mysql\_username} -p${mysql\_password} -S /data/${mysql\_port}/mysql.sock shutdown

}

function\_restart\_mysql()

{

printf "Restarting MySQL...\n"

function\_stop\_mysql

function\_start\_mysql

}

function\_kill\_mysql()

{

kill -9 $(ps -ef | grep 'bin/mysqld\_safe' | grep ${mysql\_port} | awk '{printf $2}')

kill -9 $(ps -ef | grep 'libexec/mysqld' | grep ${mysql\_port} | awk '{printf $2}')

}

case $1 in

start)

function\_start\_mysql;;

stop)

function\_stop\_mysql;;

kill)

function\_kill\_mysql;;

restart)

function\_stop\_mysql

function\_start\_mysql;;

\*)

echo "Usage: /data/${mysql\_port}/mysql {start|stop|restart|kill}";;

esac

3307 实例同上，只需要改个端口号就行了



[root@mysql-db-server 3306]# chown -R mysql.mysql /data

[root@mysql-db-server 3306]# find /data -type f -name "mysql"|xargs chmod 700

配置环境变量：

export PATH=$PATH: /redis3/bin:/apps/mysql/bin

也可以下面：



[root@mysql-db-server 3306]# which mysql

/apps/mysql/bin/mysql

#### 数据初始化

[root@mysql-db-server mysql]# cd /apps/mysql/scripts/

[root@mysql-db-server scripts]# ./mysql\_install\_db --basedir=/apps/mysql/ --datadir=/data/3306/data/ --user=mysql

[root@mysql-db-server scripts]# ./mysql\_install\_db --basedir=/apps/mysql/ --datadir=/data/3307/data/ --user=mysql

#### 启动数据库

[root@mysql-db-server data]# /data/3306/mysql start

[root@mysql-db-server data]# /data/3307/mysql start



登录



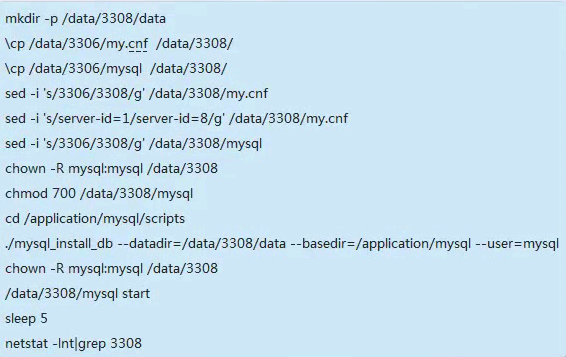
修改密码：

[root@mysql-db-server data]# mysqladmin -uroot password root -S /data/3306/mysql.sock

[root@mysql-db-server data]# [root@mysql-db-server data]# mysqladmin -uroot password root -S /data/3306/mysql.sock

[root@mysql-db-server data]# mysqladmin -uroot password root -S /data/3306]/mysql.sock mysqladmin -uroot password root -S /data/3306]/mysql.sock

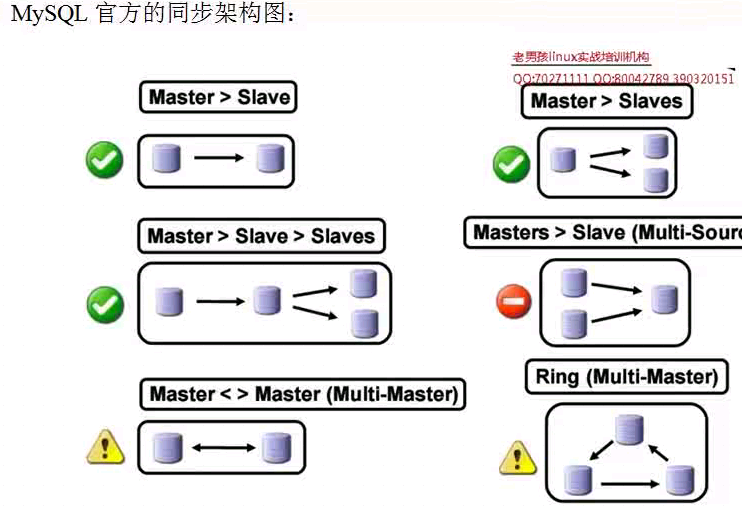
#### 创建另一个实例

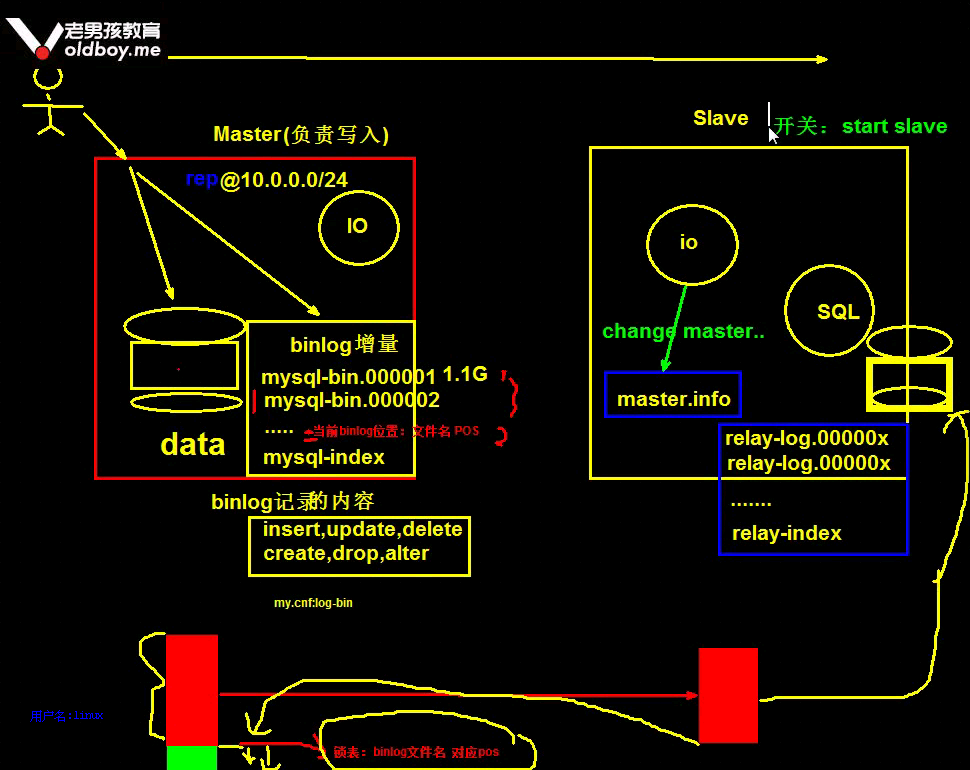
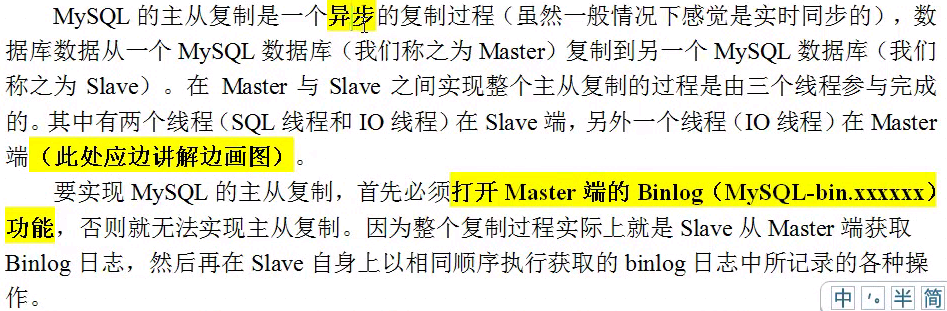


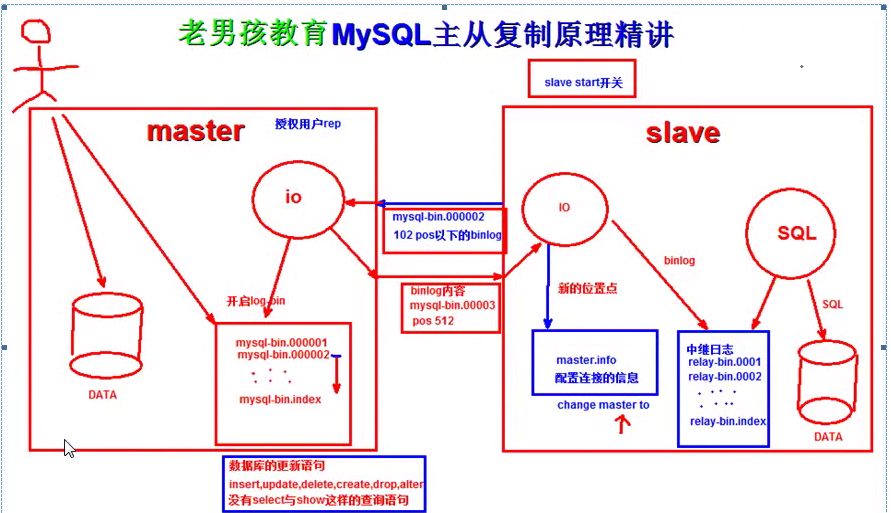




# mysql主从复制



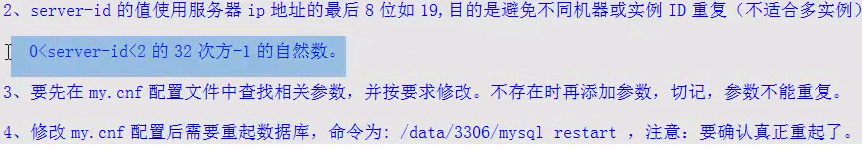




## 主从配置实践

* 1. 配置需要的服务器角色： 3306 ---》 3307





检查配置后的结果：

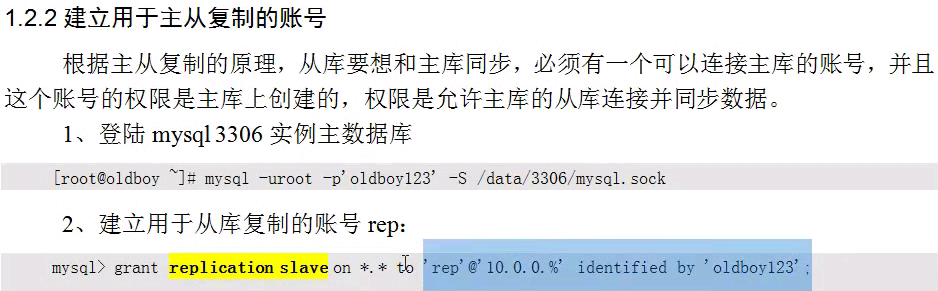
[root@mysql-db-server ~]# egrep "log-bin|server-id" /data/{3306,3307}/my.cnf

/data/3306/my.cnf:server-id = 1

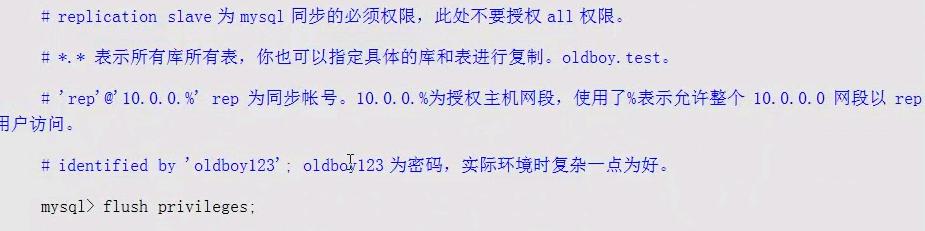
/data/3306/my.cnf:log-bin = /data/3306/mysql-bin

/data/3307/my.cnf:server-id = 3

/data/3307/my.cnf:#log-bin = /data/3307/mysql-bin

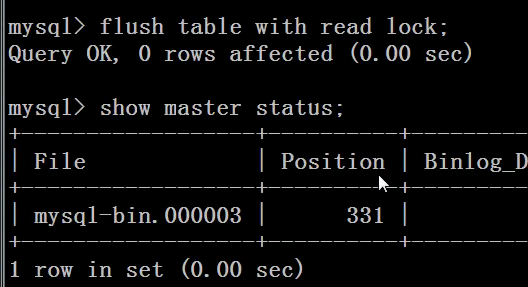


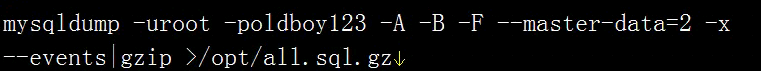
mysql> grant replication slave on \*.\* to rep@'%' identified by 'rep';



## 备份主库数据用于同步到从库

### 2.1锁表后备份





[root@mysql-db-server ~]# mysqldump -uroot -proot --events -S /data/3306/mysql.sock -A -B|zip > /opt/mysql3306\_$(date +%F).sql.gz

备份完成后比较position是否和备份前一致：

mysql> show master status;

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

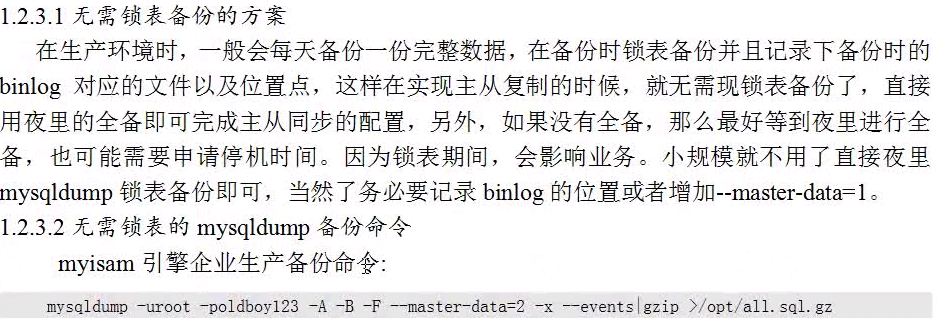
| File | Position | Binlog\_Do\_DB | Binlog\_Ignore\_DB |

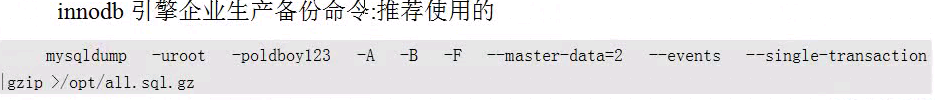
+------------------+----------+--------------+------------------+

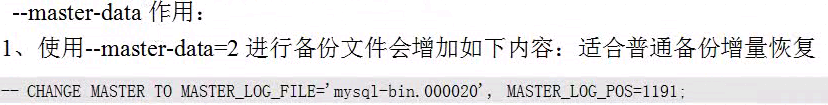
| mysql-bin.000001 | 318 | | |

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

1 row in set (0.01 sec)

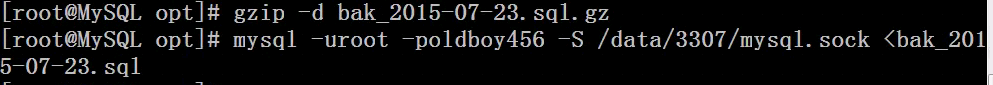






master-data = 1 生成的就不是注释了。

## 在从库上还原数据





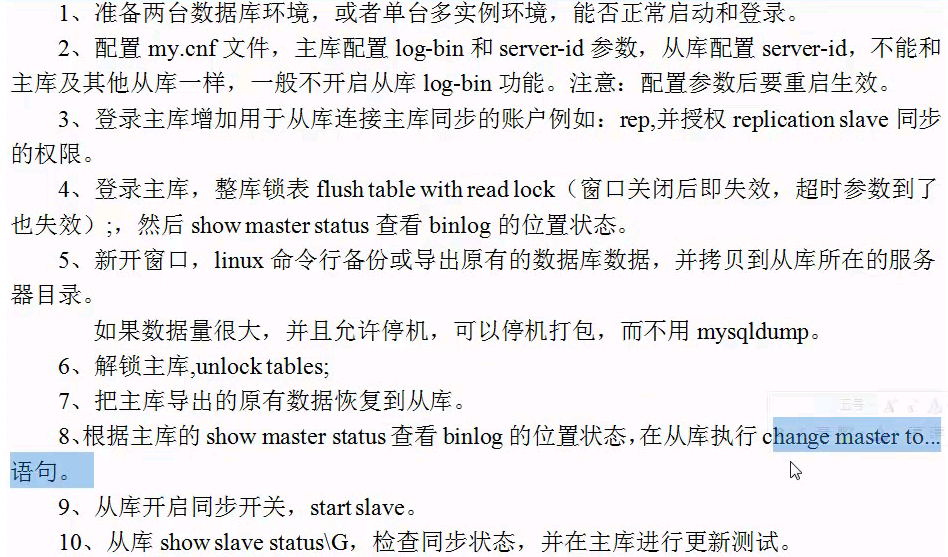
从命令行中执行：将以下信息生成到master.info当中



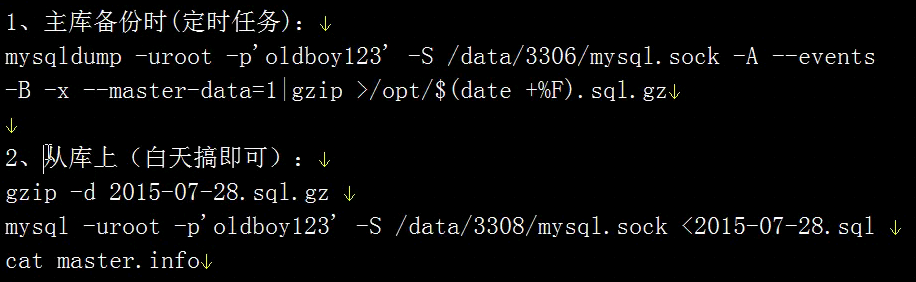
mysql > start slave;

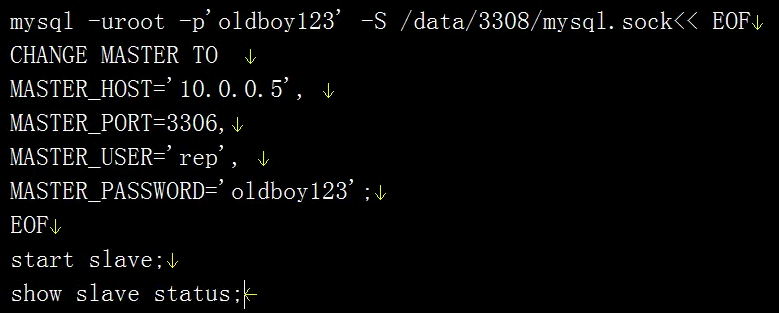
mysql > show slave status\G;





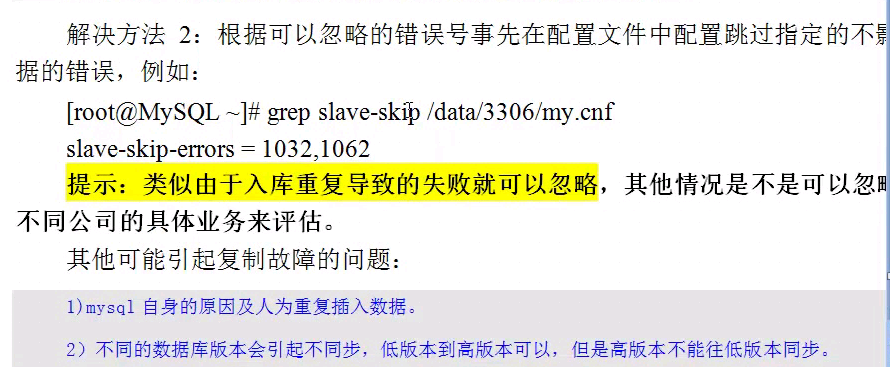




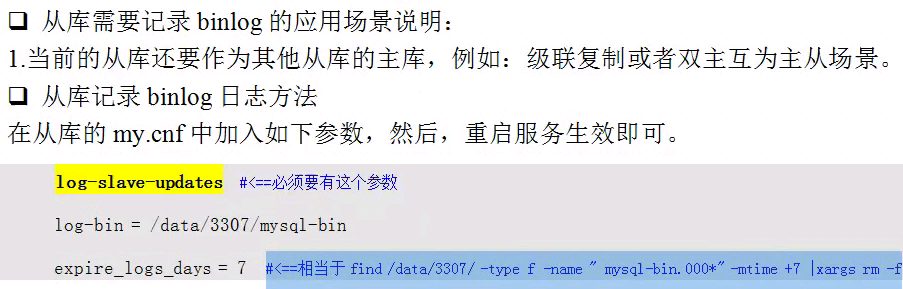


## 常见错误



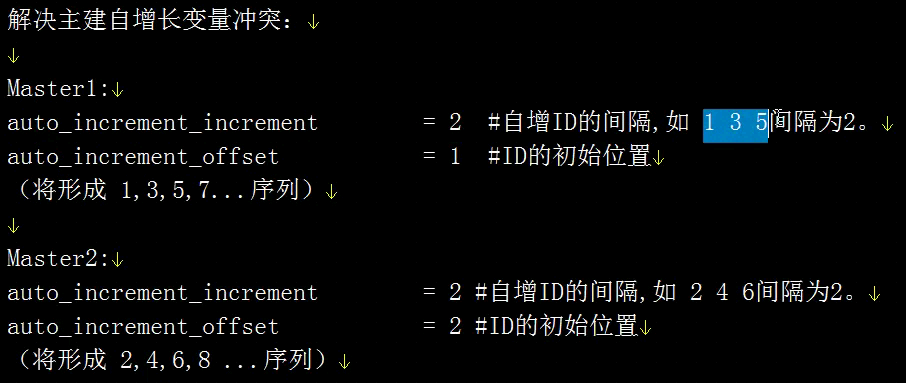


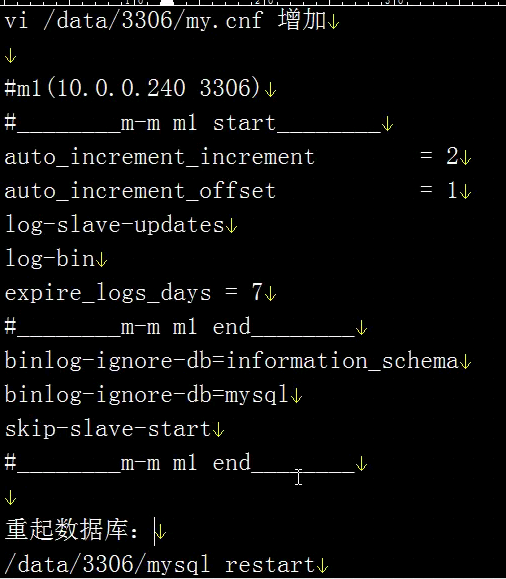
## 从库打开binlog

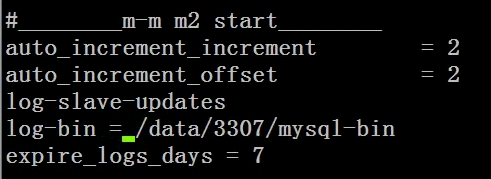


# 双主复制

双写：

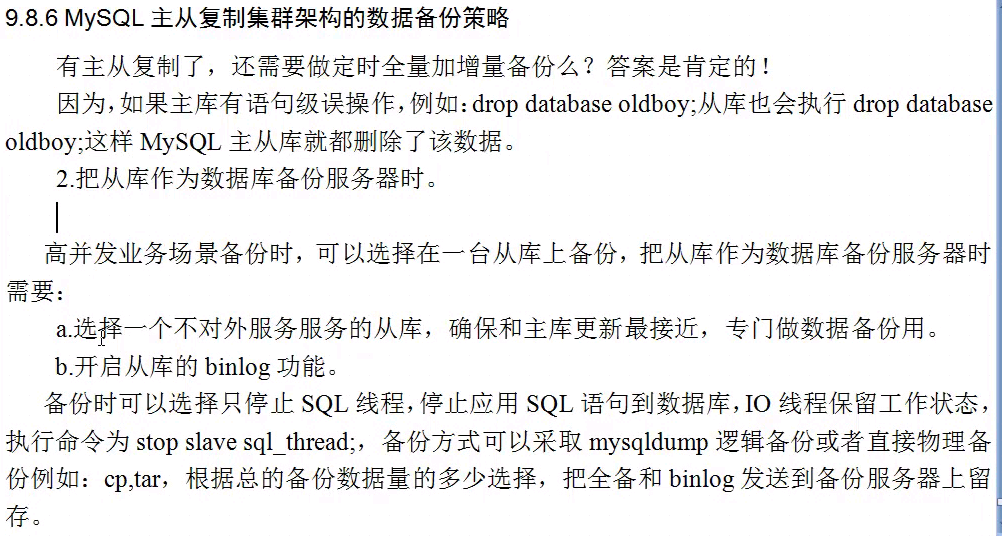


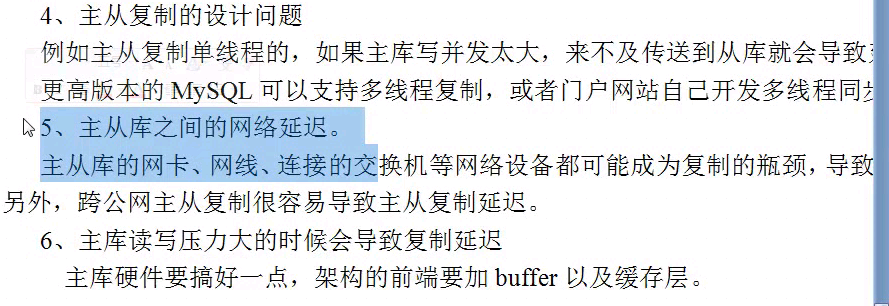
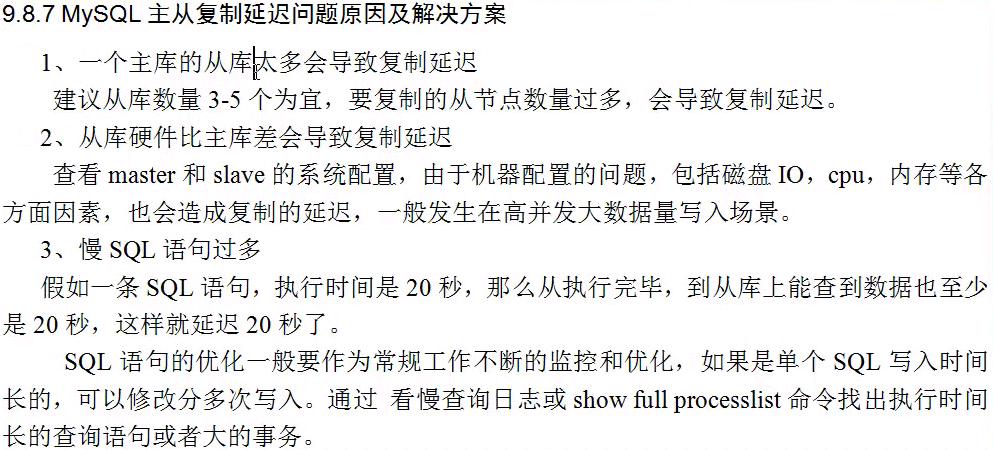




show varilables like ‘log\_%’;

# 备份策略





关于不同用户的在主从库上的授权问题：

