mysql利用binlog恢复数据操作记录

一、查看binlog信息 ：

1）查看binlog是否开启 ：

```
show variables like 'log_%';
```

2）查看所有binlog日志列表 ：

```
show master logs;
```

3）查看master状态，即最后（最新）一个binlog日志的编号名称，及其最后一个操作时间position结束点值 ：

```
show master status;
```

4）刷新日志，自刺客开始产生一个新编号的binlog日志文件 ：

```
flush logs;
show master logs;
```

每当mysqld服务重启时，会自动执行此命令，刷新binlog日志；

在mysqldump备份数据时加 -F 选项也会刷新binlog日志

5）重置（清空）所有binlog日志 ：

```
reset master;
show master logs;
```

二、查看binlog日志内容

1. 使用mysqlbinlog自带命令

在MySQL5.5以下版本使用mysqlbinlog命令时如果报错，就加上 “--no-defaults”选项

eg ：

```
mysqlbinlog mysql-bin.000002
```

这种办法读取出binlog日志的全文内容比较多，不容易分辨查看到pos点信息

2. mysql命令行查看binlog日志

基本格式 ：

```
show binlog events [IN 'log_name'] [FROM pos] [LIMIT [offset,] row_count];
```

参数解释：

IN 'log_name' ：指定要查询的binlog文件名(不指定就是第一个binlog文件)

FROM pos ：指定从哪个pos起始点开始查起(不指定就是从整个文件首个pos点开始算)

LIMIT [offset,] ：偏移量(不指定就是0)

row_count ：查询总条数(不指定就是所有行)

eg ：

查询第一个（最早）的binlog ：

```
show binlog events\G;
```

查询mysql-bin.000002这个文件 ：

```
show binlog events in 'mysql-bin.000002'\G;
```

查询mysql-bin.000002这个文件， 从pos点：624开始查起 ：

```
show binlog events in 'mysql-bin.000002' from 624\G;
```

查询mysql-bin.000002这个文件， 从pos点：624开始查起，查询10条 ：

```
shwo binlog events in 'mysql-bin.000002' from 624 limit 10;
```

查询mysql-bin.000002这个文件， 从pos点：624开始查起，偏移2行（即中间跳过2个）查询10条

```
shwo binlog events in 'mysql-bin.000002' from 624 limit 2, 10\G;
```

三、利用binlog恢复mysql数据

所谓恢复，就是让mysql将保存在binlog日志中指定段落区间的sql语句逐个重新执行一次而已。

基本格式 ：

```
mysqlbinlog mysql-bin.0000xx | mysql -u root -p dbname
```

常用参数选项解释：

--start-position=875 起始pos点

--stop-position=954 结束pos点

--start-datetime="2016-9-25 22:01:08" 起始时间点

--stop-datetime="2019-9-25 22:09:46" 结束时间点

--database=zyyshop 指定只恢复zyyshop数据库(一台主机上往往有多个数据库，只限本地log日志)

不常用选项：

-u --user=name 连接到远程主机的用户名

-p --password[=name] 连接到远程主机的密码

-h --host=name 从远程主机上获取binlog日志

--read-from-remote-server 从某个MySQL服务器上读取binlog日志

实际是将读出的binlog日志内容，通过管道符传递给mysql命令。这些命令、文件尽量写成绝对路径

场景模拟 ：

1. 凌晨4点做一个mysqldump全备，下午6点drop database ops; 中间数据有增删改。

2. 查看并备份最后一个binlog日志，并记录下关键的pos点，搞清楚到底是哪个pos点的操作导致了数据库的破坏。

```
show master status;
cp -v mysql-bin.000003 /opt/backup/
```

3. 刷新日志索引，重新开始新的binlog日志记录文件。

```
flush logs;
show master status;
```

4. 读取binlog日志，分析问题 ：

```
mysqlbinlog mysql-bin.000003
or
show binlog events in 'mysql-bin.000003';
or
show binlog events in 'mysql-bin.000003'\G;
```

5. 先把凌晨4点全备的数据恢复 ：

```
gzip -d ops_2016-09-25.sql.gz
mysql -u root -p -v < ops_2016-09-25.sql.gz
```

6. 从binlog恢复数据

1） 完全恢复（需要手动vi编辑mysql-bin.000003, 将那条drop 语句剔除掉）

```
/usr/bin/mysqlbinlog /var/lib/mysql/mysql-bin.000003 | /usr/bin/mysql -uroot -p123456 -v ops
```

2）指定pos结束点恢复（部分恢复）

--stop-position=471 pos结束节点（按照事务区间算，是471）

```
/usr/bin/mysqlbinlog --stop-position=471 --database=ops /var/lib/mysql/mysql-bin.000003 | /usr/bin/mysql -u root -p -v ops
```

3）指定pos点区间恢复(部分恢复)

单独恢复name='李四' 这步操作 ：

```
/usr/bin/mysqlbinlog --start-position=538 --stop-position=646 --database=ops /var/lib/mysql/mysql-bin.000003 | /usr/bin/mysql -u root -p -v ops
```

按照失误区间单独恢复name=‘李四’这个事物 ：

```
/usr/bin/mysqlbinlog --start-position=471 --stop-position=673 --database=ops /var/lib/mysql/mysql-bin.000003 | /usr/bin/mysql -u root -p -v ops
```

单独恢复name='小二‘这步操作 ：

按照binlog日志区间单独恢复 ：

```
/usr/bin/mysqlbinlog --start-position=740 --stop-position=848 --database=ops /var/lib/mysql/mysql-bin.000003 | /usr/bin/mysql -u root -p -v ops
```

按照事物区间单独恢复 ：

```
/usr/bin/mysqlbinlog --start-position=673 --stop-position=875 --database=ops /var/lib/mysql/mysql-bin.000003 | /usr/bin/mysql -u root -p -v ops
```

将name=‘李四’，name='小二‘多步操作一起恢复，需要按事物区间 ：

```
/usr/bin/mysqlbinlog --start-position=471 --stop-position=875 --database=ops /var/lib/mysql/mysql-bin.000003 | /usr/bin/mysql -u root -p -v ops
```

查看数据库 ：

```
select * from member;
```

另外：

也可指定时间节点区间恢复(部分恢复)：

除了用pos节点的办法进行恢复，也可以通过指定时间节点区间进行恢复，按时间恢复需要用mysqlbinlog命令读取binlog日志内容，找时间节点。

如上，误删除ops库后：

先进行全备份恢复

```
[root@vm-002 backup]# mysql -uroot -p -v < ops_2016-09-25.sql
```

查看ops数据库

查看mysq-bin00003日志，找出时间节点

恢复到更改“name='李四'”之前的数据

```
[root@vm-002 ~]# /usr/bin/mysqlbinlog --start-datetime="2016-09-25 21:57:19″ --stop-datetime="2016-09-25 21:58:41″ --database=ops /var/lib/mysql/mysql-bin.000003 | /usr/bin/mysql -uroot -p123456 -v ops
[root@vm-002 ~]# /usr/bin/mysqlbinlog --start-datetime="2016-09-25 21:58:41″ --stop-datetime="2016-09-25 21:58:56″ --database=ops
```

```
/var/lib/mysql/mysql-bin.000003 | /usr/bin/mysql -uroot -p123456 -v ops
[root@vm-002 ~]# /usr/bin/mysqlbinlog --start-datetime="2016-09-25 21:58:56″ --stop-datetime="2016-09-25 22:01:08″ --database=ops
/var/lib/mysql/mysql-bin.000003 | /usr/bin/mysql -uroot -p123456 -v ops
```

```
mysql> show binlog events in 'mysql-bin.000003';
+------------------+-----+-------------+-----------+-------------+----------------------------------------------------------------
-------------------------------------------------------------+
| Log_name | Pos | Event_type | Server_id | End_log_pos | Info |
+------------------+-----+-------------+-----------+-------------+----------------------------------------------------------------
-------------------------------------------------------------+
| mysql-bin.000003 | 4 | Format_desc | 1 | 106 | Server ver: 5.1.73-log, Binlog ver: 4 |
| mysql-bin.000003 | 106 | Query | 1 | 173 | BEGIN |
| mysql-bin.000003 | 173 | Intvar | 1 | 201 | INSERT_ID=3 |
| mysql-bin.000003 | 201 | Query | 1 | 444 | use `ops`; insert into ops.member(`name`,`sex`,`age`,`gsan','w',21,'cls5'),
('lisi','m',20,'cls4'),('wangwu','w',26,'cls6') |
| mysql-bin.000003 | 444 | Xid | 1 | 471 | COMMIT /* xid=66 */ |
| mysql-bin.000003 | 471 | Query | 1 | 538 | BEGIN |
| mysql-bin.000003 | 538 | Query | 1 | 646 | use `ops`; update ops.member set name='李四' where id= |
| mysql-bin.000003 | 646 | Xid | 1 | 673 | COMMIT /* xid=68 */ |
| mysql-bin.000003 | 673 | Query | 1 | 740 | BEGIN |
| mysql-bin.000003 | 740 | Query | 1 | 848 | use `ops`; update ops.member set name='小二' where id= |
| mysql-bin.000003 | 848 | Xid | 1 | 875 | COMMIT /* xid=69 */ |
| mysql-bin.000003 | 875 | Query | 1 | 954 | drop database ops |
| mysql-bin.000003 | 954 | Rotate | 1 | 997 | mysql-bin.000004;pos=4 |
+------------------+-----+-------------+-----------+-------------+----------------------------------------------------------------
-------------------------------------------------------------+
13 rows in set (0.00 sec)
```

参考 ： http://www.cnblogs.com/kevingrace/p/5907254.html

MySQL binlog help详细信息 ：

```
[root@localhost binlog]# mysqlbinlog --help
mysqlbinlog Ver 3.3 for redhat-linux-gnu at x86_64
Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Dumps a MySQL binary log in a format usable for viewing or for piping to
the mysql command line client.

Usage: mysqlbinlog [options] log-files
  -?, --help          Display this help and exit.
  --base64-output[=name]
                      Determine when the output statements should be
                      base64-encoded BINLOG statements: 'never' disables it and
                      works only for binlogs without row-based events;
                      'decode-rows' decodes row events into commented SQL
                      statements if the --verbose option is also given; 'auto'
                      prints base64 only when necessary (i.e., for row-based
                      events and format description events); 'always' prints
```

```
                    base64 whenever possible. ’always’ is for debugging only
                    and should not be used in a production system. If this
                    argument is not given, the default is ’auto’; if it is
                    given with no argument, ’always’ is used.
--character-sets-dir=name
                    Directory for character set files.
-d, --database=name List entries for just this database (local log only).
--debug-check       Check memory and open file usage at exit .
--debug-info        Print some debug info at exit.
-D, --disable-log-bin
                    Disable binary log. This is useful, if you enabled
                    --to-last-log and are sending the output to the same
                    MySQL server. This way you could avoid an endless loop.
                    You would also like to use it when restoring after a
                    crash to avoid duplication of the statements you already
                    have. NOTE: you will need a SUPER privilege to use this
                    option.
-F, --force-if-open Force if binlog was not closed properly.
-f, --force-read    Force reading unknown binlog events.
-H, --hexdump       Augment output with hexadecimal and ASCII event dump.
-h, --host=name     Get the binlog from server.
-l, --local-load=name
                    Prepare local temporary files for LOAD DATA INFILE in the
                    specified directory.
-o, --offset=#      Skip the first N entries.
-p, --password[=name]
                    Password to connect to remote server.
-P, --port=#        Port number to use for connection or 0 for default to, in
                    order of preference, my.cnf, $MYSQL_TCP_PORT,
                    /etc/services, built-in default (3306).
--position=#        Deprecated. Use --start-position instead.
--protocol=name     The protocol to use for connection (tcp, socket, pipe,
                    memory).
-R, --read-from-remote-server
                    Read binary logs from a MySQL server.
-r, --result-file=name
                    Direct output to a given file.
--server-id=#       Extract only binlog entries created by the server having
                    the given id.
--set-charset=name  Add ’SET NAMES character_set’ to the output.
-s, --short-form    Just show regular queries: no extra info and no row-based
                    events. This is for testing only, and should not be used
                    in production systems. If you want to suppress
                    base64-output, consider using --base64-output=never
                    instead.
-S, --socket=name   The socket file to use for connection.
--start-datetime=name
                    Start reading the binlog at first event having a datetime
                    equal or posterior to the argument; the argument must be
                    a date and time in the local time zone, in any format
                    accepted by the MySQL server for DATETIME and TIMESTAMP
                    types, for example: 2004-12-25 11:25:56 (you should
                    probably use quotes for your shell to set it properly).
-j, --start-position=#
                    Start reading the binlog at position N. Applies to the
                    first binlog passed on the command line.
--stop-datetime=name
                    Stop reading the binlog at first event having a datetime
```

```
                      equal or posterior to the argument; the argument must be
                      a date and time in the local time zone, in any format
                      accepted by the MySQL server for DATETIME and TIMESTAMP
                      types, for example: 2004-12-25 11:25:56 (you should
                      probably use quotes for your shell to set it properly).
  --stop-position=#   Stop reading the binlog at position N. Applies to the
                      last binlog passed on the command line.
  -t, --to-last-log   Requires -R. Will not stop at the end of the requested
                      binlog but rather continue printing until the end of the
                      last binlog of the MySQL server. If you send the output
                      to the same MySQL server, that may lead to an endless
                      loop.
  -u, --user=name     Connect to the remote server as username.
  -v, --verbose       Reconstruct SQL statements out of row events. -v -v adds
                      comments on column data types.
  -V, --version       Print version and exit.
  --open_files_limit=#
                      Used to reserve file descriptors for use by this program.

Variables (--variable-name=value)
and boolean options {FALSE|TRUE}  Value (after reading options)
--------------------------------- -----------------------------
base64-output                     (No default value)
character-sets-dir                (No default value)
database                          (No default value)
debug-check                       FALSE
debug-info                        FALSE
disable-log-bin                   FALSE
force-if-open                     TRUE
force-read                        FALSE
hexdump                           FALSE
host                              (No default value)
local-load                        (No default value)
offset                            0
port                              0
position                          4
read-from-remote-server           FALSE
server-id                         0
set-charset                       (No default value)
short-form                        FALSE
socket                            (No default value)
start-datetime                    (No default value)
start-position                    4
stop-datetime                     (No default value)
stop-position                     18446744073709551615
to-last-log                       FALSE
user                              (No default value)
open_files_limit                  64
[root@localhost binlog]#
```

eg :

```
mysqlbinlog  ’/opt/mysql/data/mysql-bin.000008’ --start-position=1036 --stop-position=1447 |mysql -uroot -p
```