1、参考网站：<https://www.linuxidc.com/Linux/2015-07/120296.htm>

2、简介

Xtrabackup是由percona提供的mysql数据库备份工具，据官方介绍，这也是世界上惟一一款开源的能够对innodb和xtradb数据库进行热备的工具。特点：

(1)备份过程快速、可靠；

(2)备份过程不会打断正在执行的事务；

(3)能够基于压缩等功能节约磁盘空间和流量；

(4)自动实现备份检验；

(5)还原速度快；

2、安装

yum -y install perl-DBD-MySQL

wget https://www.percona.com/downloads/XtraBackup/XtraBackup-2.2.8/binary/redhat/7/x86\_64/percona-xtrabackup-2.2.8-5059.el7.x86\_64.rpm

 rpm -ivh percona-xtrabackup-2.2.3-4982.el6.x86\_64.rpm

rpm -ql xtrabackup

3、备份说明

使用innobakupex备份时，其会调用xtrabackup备份所有的InnoDB表，复制所有关于表结构定义的相关文件(.frm)、以及MyISAM、MERGE、CSV和ARCHIVE表的相关文件，同时还会备份触发器和数据库配置信息相关的文件。这些文件会被保存至一个以时间命令的目录中。

在使用innobackupex进行备份时，还可以使用--no-timestamp选项来阻止命令自动创建一个以时间命名的目录；如此一来，innobackupex命令将会创建一个BACKUP-DIR目录来存储备份数据。

4、完全备份

**一定要开启bin-log**

[root@redhat7 2018-03-10\_21-37-39]# cat /etc/my.cnf

[mysqld]

datadir=/var/lib/mysql

socket=/var/lib/mysql/mysql.sock

innodb\_log\_file\_size=5M

log-bin=/var/lib/mysql/mysql-bin

# Disabling symbolic-links is recommended to prevent assorted security risks

symbolic-links=0

# Settings user and group are ignored when systemd is used.

# If you need to run mysqld under a different user or group,

# customize your systemd unit file for mariadb according to the

# instructions in http://fedoraproject.org/wiki/Systemd

[mysqld\_safe]

log-error=/var/log/mariadb/mariadb.log

pid-file=/var/run/mariadb/mariadb.pid

#

# include all files from the config directory

#

!includedir /etc/my.cnf.d

[root@redhat7 2018-03-10\_21-37-39]#

innobackupex --user=root --password=123456 /backup/

备份时报了一个错

innobackupex:Error: The xtrabackup child process has died at /usr/bin/innobackupex line2672.

解决方法：修改innodb日志文件大小为5M，修改完成重启服务生效

[root@redhat7 2018-03-10\_21-37-39]# ll

total 30748

-rw-r--r-- 1 root root 262 Mar 10 21:37 backup-my.cnf

-rw-r----- 1 root root 18874368 Mar 10 21:40 ibdata1

-rw-r--r-- 1 root root 5242880 Mar 10 21:40 ib\_logfile0

-rw-r--r-- 1 root root 5242880 Mar 10 21:40 ib\_logfile1

drwxr-xr-x 2 root root 4096 Mar 10 21:37 mysql

drwxr-xr-x 2 root root 4096 Mar 10 21:37 performance\_schema

drwxr-xr-x 2 root root 4096 Mar 10 21:37 test

-rw-r--r-- 1 root root 21 Mar 10 21:37 xtrabackup\_binlog\_info

-rw-r----- 1 root root 89 Mar 10 21:40 xtrabackup\_checkpoints

-rw-r--r-- 1 root root 559 Mar 10 21:37 xtrabackup\_info

-rw-r----- 1 root root 2097152 Mar 10 21:40 xtrabackup\_logfile

[root@redhat7 2018-03-10\_21-37-39]#

5、文件说明

文件说明：

(1)xtrabackup\_checkpoints—— 备份类型（如完全或增量）、备份状态（如是否已经为prepared状态）和LSN(日志序列号)范围信息；

每个InnoDB页(通常为16k大小)都会包含一个日志序列号，即LSN。LSN是整个数据库系统的系统版本号，每个页面相关的LSN能够表明此页面最近是如何发生改变的。

[root@MariaDB 2015-06-15\_23-49-43]# catxtrabackup\_checkpoints

backup\_type = full-backuped  #备份类型，full-backuped表示完全备份

from\_lsn = 0                    #日志序列开始

to\_lsn = 1597945                #最大日志序列号

last\_lsn = 1597945              #当前日志序列号

compact = 0                      #表示没有打包

(2)xtrabackup\_binlog\_info —— mysql服务器当前正在使用的二进制日志文件及至备份这一刻为止二进制日志事件的位置。

[root@MariaDB 2015-06-15\_23-49-43]# catxtrabackup\_binlog\_info

mysql-bin.000005  245

(3)backup-my.cnf —— 备份命令用到的配置选项信息和备份无关的不会记录，备份配置文件的话需要单独备份

[root@MariaDB 2015-06-15\_23-49-43]# catbackup-my.cnf

# This MySQL options file was generated byinnobackupex.

# The MySQL server

[mysqld]

innodb\_checksum\_algorithm=innodb

innodb\_log\_checksum\_algorithm=innodb

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

innodb\_log\_files\_in\_group=2

innodb\_log\_file\_size=5242880

innodb\_fast\_checksum=0

innodb\_page\_size=16384

innodb\_log\_block\_size=512

innodb\_undo\_tablespaces=0

 （4）xtrabackup\_info —— 记录了mariadb的版本信息和一些属性信息，还原是检测版本匹配度时用到

[root@MariaDB 2015-06-15\_23-49-43]# catxtrabackup\_info

uuid = 25b79086-1376-11e5-980a-000c29bad792

name =

tool\_name = innobackupex

tool\_command = --user=root --password=... /backup/

tool\_version = 1.5.1-xtrabackup

ibbackup\_version = xtrabackup version 2.2.3 basedon MySQL server 5.6.17 Linux (x86\_64) (revision id: )

server\_version = 5.5.43-MariaDB-log

start\_time = 2015-06-15 23:49:43

end\_time = 2015-06-15 23:49:46

lock\_time = 1

binlog\_pos = filename 'mysql-bin.000005', position245

innodb\_from\_lsn = 0

innodb\_to\_lsn = 1597945

partial = N

incremental = N

format = file

compact = N

compressed = N

encrypted = N

6、准备一个完全备份

一般情况下，在备份完成后，数据尚且不能用于恢复操作，因为备份的数据中可能会包含尚未提交的事务或已经提交但尚未同步至数据文件中的事务。因此，此时数据文件仍处于不一致状态。“准备”的主要作用正是通过回滚未提交的事务及同步已经提交的事务至数据文件也使得数据文件处于一致性状态。

innobakupex命令的--apply-log选项可用于实现上述功能。如下面的命令：

[root@MariaDB 2015-06-15\_23-49-43]# innobackupex--apply-log /backup/2015-06-15\_23-49-43/

如果执行正确，其最后输出的一行信息通常如下：

150615 23:58:59 innobackupex: completed OK!

在实现“准备”的过程中，innobackupex通常还可以使用--use-memory选项来指定其可以使用的内存的大小，默认通常为100M。如果有足够的内存可用，可以多划分一些内存给prepare的过程，以提高其完成速度。

3、从一个完全备份中恢复数据

innobackupex命令的--copy-back选项用于执行恢复操作，其通过复制所有数据相关的文件至mysql服务器DATADIR目录中来执行恢复过程。innobackupex通过backup-my.cnf来获取DATADIR目录的相关信息。

语法：

# innobackupex --copy-back  /path/to/BACKUP-DIR

当数据恢复至DATADIR目录以后，还需要确保所有数据文件的属主和属组均为正确的用户，如mysql，否则，在启动mysqld之前还需要事先修改数据文件的属主和属组。

模拟数据库损坏

删除mariadb数据目录中的所有内容，模拟数据库损坏

[root@MariaDB~]# rm -rf /mydata/data/\*

[root@MariaDB~]# ll /mydata/data/

total0

恢复数据

[root@MariaDB~]# innobackupex --copy-back /backup/2015-06-15\_23-49-43/

[root@MariaDB ~]# chown -R  mysql:mysql /mydata/data/

[root@MariaDB~]# ll /mydata/data/

total 28692

drwxr-xr-x2 mysql mysql    4096 Jun 16 00:01hellodb

-rw-r--r--1 mysql mysql 18874368 Jun 16 00:01 ibdata1

-rw-r--r--1 mysql mysql  5242880 Jun 16 00:01ib\_logfile0

-rw-r--r--1 mysql mysql  5242880 Jun 16 00:01ib\_logfile1

drwxr-xr-x2 mysql mysql    4096 Jun 16 00:01 mysql

drwxr-xr-x2 mysql mysql    4096 Jun 16 00:01performance\_schema

drwxr-xr-x2 mysql mysql    4096 Jun 16 00:01 test

-rw-r--r--1 mysql mysql      559 Jun 16 00:01xtrabackup\_info

恢复完成之后，还需要启动一下Mysql服务，否则Mariadb是是不会记录二进制日志的，这个时候启动起来是重新记录二进制日志的。

[root@MariaDB ~]# service mysqld start

提示：在恢复完成之后应该再次做一次完全备份，后期的增量备份都依照这次的完全备份来做。

[root@MariaDB ~]# innobackupex --user root --password centos/backup

7、第一次增量备份

使用innobackupex进行增量备份

每个InnoDB的页面都会包含一个LSN信息，每当相关的数据发生改变，相关的页面的LSN就会自动增长。这正是InnoDB表可以进行增量备份的基础，即innobackupex通过备份上次完全备份之后发生改变的页面来实现。

要实现第一次增量备份，可以使用下面的命令进行：

# innobackupex --incremental /backup--incremental-basedir=BASEDIR

其中，BASEDIR指的是完全备份所在的目录，此命令执行结束后，innobackupex命令会在/backup目录中创建一个新的���时间命名的目录以存放所有的增量备份数据。另外，在执行过增量备份之后再一次进行增量备份时，其--incremental-basedir应该指向上一次的增量备份所在的目录。

需要注意的是，增量备份仅能应用于InnoDB或XtraDB表，对于MyISAM表而言，执行增量备份时其实进行的是完全备份。

“准备”(prepare)增量备份与整理完全备份有着一些不同，尤其要注意的是：

(1)需要在每个备份(包括完全和各个增量备份)上，将已经提交的事务进行“重放”。“重放”之后，所有的备份数据将合并到完全备份上。

(2)基于所有的备份将未提交的事务进行“回滚”。

于是，操作就变成了：

# innobackupex --apply-log --redo-onlyBASE-DIR

接着执行：

# innobackupex --apply-log --redo-onlyBASE-DIR --incremental-dir=INCREMENTAL-DIR-1

而后是第二个增量：

# innobackupex --apply-log --redo-only BASE-DIR--incremental-dir=INCREMENTAL-DIR-2

其中BASE-DIR指的是完全备份所在的目录，而INCREMENTAL-DIR-1指的是第一次增量备份的目录，INCREMENTAL-DIR-2指的是第二次增量备份的目录，其它依次类推，即如果有多次增量备份，每一次都要执行如上操作；

备份过程

由于上面的完全备份恢复之后又进行了一次完全备份，这里不再进行完全备份，增量备份直接根据上面的完全备份进行。

查看lsn日志的起始和结束位置，第一次增量备份的LSN起始位置就是完全备份LSN的结束位置

[root@MariaDB ~]# cat/backup/2015-06-15\_23-45-01/xtrabackup\_checkpoints

backup\_type = full-backuped

from\_lsn = 0

to\_lsn = 1602246

last\_lsn = 1602246

compact = 0

修改数据库内容

MariaDB[hellodb]> select \* from tb1;

+------+

|id  |

+------+

|    1 |

|    2 |

|    3 |

|  21 |

|  22 |

|  23 |

+------+

MariaDB[hellodb]> delete from tb1 where id=21;

MariaDB[hellodb]> delete from tb1 where id=22;

MariaDB[hellodb]> select \* from tb1;

+------+

|id  |

+------+

|    1 |

|    2 |

|    3 |

|  23 |

+------+

第一次增量备份

只需要指向上一次完全备份的位置和这一次增量备份的存储位置即可

[root@MariaDB ~]# innobackupex --user root--password centos --incremental /backup/ --incremental-basedir=/backup/2015-06-15\_23-45-01/

备份完成之后查看checkpoints信息

[root@MariaDB ~]# cat/backup/2015-06-15\_23-47-02/xtrabackup\_checkpoints

backup\_type = incremental  #备份类型表示为增量

from\_lsn = 1602246          #表示备份的起始位置

to\_lsn = 1604539

last\_lsn = 1604539

compact = 0

再次修改一些数据

MariaDB[hellodb]> insert into tb1 values (1000),(9000);

MariaDB[hellodb]> select \* from tb1;

+------+

|id  |

+------+

|    1 |

|    2 |

|    3 |

|  23 |

|1000 |

|9000 |

+------+

第二次做增量备份

这是备份是基于第一次增量备份进行

[root@MariaDB ~]# innobackupex --user root--password centos --incremental /backup--incremental-basedir=/backup/2015-06-15\_23-47-02/

[root@MariaDB ~]# cat/backup/2015-06-15\_23-51-55/xtrabackup\_checkpoints

backup\_type = incremental

from\_lsn = 1604539 #备份起始位置是第一次增量备份的结束位置

to\_lsn = 1604539

last\_lsn = 1605623

compact = 0

再次修改一次数据，这次修改没有做备份

MariaDB[hellodb]> insert into tb1 values (88888);

MariaDB[hellodb]> select \* from tb1;

+-------+

|id    |

+-------+

|    1 |

|    2 |

|    3 |

|    23 |

|  1000 |

|  9000 |

|88888 |

+-------+

说明：由于没有实现将bin-log文件和数据目录分开，所以我先导入二进制日志文件中的修改内容。

查看binlog\_info日志位置记录位置为1621，那么只需要把1621之后的修改导出为sql文件即可。

[root@MariaDB~]# cat /backup/2015-06-15\_23-51-55/xtrabackup\_binlog\_info

mysql-bin.000005    979

[root@MariaDB ~]# mysqlbinlog --start-position=979/mydata/data/mysql-bin.000005 > /backup/timepoint.sql

模拟损坏

停止mysqld服务，然后删除数据目录下的所有数据，模拟硬盘损坏

[root@MariaDB~]# service mysqld stop

[root@MariaDB~]# rm -rf /mydata/data/\*

数据恢复

恢复过程：合并完全备份

[root@MariaDB ~]# innobackupex --apply-log --redo-only /backup/2015-06-15\_23-45-01/

合并第一个增量备份

[root@MariaDB ~]# innobackupex --apply-log --redo-only /backup/2015-06-15\_23-45-01/--incremental-dir=/backup/2015-06-15\_23-47-02/

合并第二个增量备份

[root@MariaDB ~]# innobackupex --apply-log --redo-only /backup/2015-06-15\_23-45-01/--incremental-dir=/backup/2015-06-15\_23-51-55/

使用合并后的命令进行恢复

[root@MariaDB ~]# innobackupex --copy-back /backup/2015-06-15\_23-45-01/

设置完成之后修改数据目录的属主属组

[root@MariaDB data]# chown -R mysql.mysql ./\*

[root@MariaDB data]# ll

total 18452

drwxr-xr-x 2 mysql mysql    4096 Jun 16 00:01 hellodb

-rw-r--r-- 1 mysql mysql 18874368 Jun 16 00:01ibdata1

drwxr-xr-x 2 mysql mysql    4096 Jun 16 00:01 mysql

drwxr-xr-x 2 mysql mysql    4096 Jun 16 00:01 performance\_schema

drwxr-xr-x 2 mysql mysql    4096 Jun 16 00:01 test

-rw-r--r-- 1 mysql mysql      632 Jun 16 00:01 xtrabackup\_info

设置完成之后启动Mysql服务器

[root@MariaDBdata]# service mysqld start

查看数据库这个时候还没有第二次增量备份之后修改的数据

MariaDB[hellodb]> select \* from tb1;

+------+

|id  |

+------+

|    1 |

|    2 |

|    3 |

|  23 |

|1000 |

|9000 |

+------+

将从二进制导出的sql文件，导入到数据库

[root@MariaDB ~]# mysql -u root -p < /backup/timepoint.sql

查看数据就已经恢复了

MariaDB[hellodb]> select \* from tb1;

+-------+

|id    |

+-------+

|    1 |

|    2 |

|    3 |

|    23 |

|  1000 |

|  9000 |

|88888 |

+-------+