一、mysql备份讲解

1.1 备份的作用

论数据的重要性

1.2 如何实现备份？都有哪些方法可以实现数据的备份？

bin-log

使用二进制日志的方式进行备份，主要实现过程通过实现mysql的主从备份即可

备注：建议将bin-log实时备份至远程服务器中，以防机器故障而导致日志丢失

mysqldump

mydumper

Xtrabackup \*

二、mysqldump

常用参数说明：

--version, -V 输出mysqldump版本信息并退出

-----------------------------------------------------------------------

--help 显示帮助信息并退出

-----------------------------------------------------------------------

--password, -p 连接数据库密码

--user, -u 连接数据库的用户名

--port, -P 连接数据库端口号

--protocol 使用的连接协议，例如：tcp udp socket

[root@node2 mysql]# mysqldump -uroot -p -B m2 --protocol=tcp

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

命令格式：

# mysqldump -uroot -p -A >/mysqlback/mysql\_all.sql

作用：

将数据库所有内容备份至/root下，并设置文件名为mysql\_all.sql

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

------------------------------------------------------------------------

--all-databases , -A 导出全部数据库(全备)

[root@node2 mysql]# mysqldump -uroot -p -A >/myback/mysql\_all\_back.sql

------------------------------------------------------------------------

--databases -B 导出指定的数据库,可指定多个

mysqldump -uroot -p -B m2 test mysql a b

------------------------------------------------------------------------

--no-create-db, -n 导出数据，但不导出库

mysqldump -uroot -p -B m2 -n

------------------------------------------------------------------------

--no-create-info, -t 仅导出数据，不导出数据结构

mysqldump -uroot -p -B m2 -t

------------------------------------------------------------------------

--no-data, -d 仅导出数据结构，不导出数据

mysqldump -uroot -p -A -d

------------------------------------------------------------------------

--add-drop-database 导入数据时，先删除原来的库

mysqldump -uroot -p -B m2 --add-drop-database

------------------------------------------------------------------------

--add-drop-table 导入数据时，先删除原来的表

mysqldump -uroot -p -B m2 --skip-add-drop-table --关闭以上选项

------------------------------------------------------------------------

--add-locks 导入数据时，先锁表，完成后自动解锁（默认开启）

mysqldump -uroot -p -B m2 --skip-add-locks --关闭以上选项

------------------------------------------------------------------------

--apply-slave-statements 导入数据时，如果备份的数据库为slave，先自动停止slave，完成后自动启动

mysqldump -uroot -p -B m2 --apply-slave-statements

------------------------------------------------------------------------

--tables 导出指定的表，须配合-B使用

mysqldump -uroot -p -B m2 --tables test

------------------------------------------------------------------------

--default-character-set 设置导出时数据的字符集，默认为utf-8

mysqldump -uroot -p -B m2 --default-character-set=latin1

------------------------------------------------------------------------

--where, -w 导出时，在后方添加条件语句，只导出符合条件的数据

mysqldump -uroot -p -B m2 --where="user='root'"

------------------------------------------------------------------------

--flush-logs , -F 开始导出之前刷新bin-log日志。

备注：如果一次导出多个库，将会多次刷新bin-log，只有在同时使用--lock-all-tables或者--master-data才不会出现类似情况

mysqldump -uroot -p -B m2 -F

------------------------------------------------------------------------

--force 在导出过程中忽略出现的SQL错误

mysqldump -uroot -p -B m2 --force

------------------------------------------------------------------------

--ignore-table 不导出指定表，如果要忽略多个表，需要使用多次该参数

例如：--ignore-table=m2.a --ignore-table=m2.b ……

mysqldump -uroot -p -B m2 --ignore-table=m2.a

------------------------------------------------------------------------

--lock-all-tables, -x 导出数据库之前，先锁定所有表，确保导出时无数据插入，保证数据一致性

mysqldump -uroot -p -B m2 --lock-all-tables

------------------------------------------------------------------------

--log-error 导出过程中，如果有错误信息，将写入到指定文件

mysqldump -uroot -p -B m2 --log-error=/root/mysqldump\_err.log

------------------------------------------------------------------------

--dump-date 导出时，在文件中添加导出时间，默认开启

mysqldump -uroot -p -B m2 --skip-dump-date

------------------------------------------------------------------------

--socket, -S 导出时指定mysql的socket文件位置，默认路径/tmp/mysql.sock

mysqldump -uroot -p -B m2 --socket=/tmp/mysqld.sock

------------------------------------------------------------------------

--xml, -X 导出时，将数据格式转换为XML

mysqldump -uroot -p -B m2 --xml

------------------------------------------------------------------------

以上备份均为全备方式，如果要实现增量备份，最简单的方法是通过mysqldump配合bin-log来实现

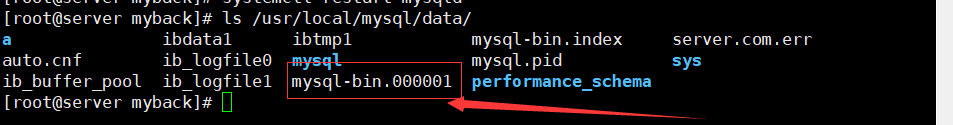
实现原理：通过mysqldump命令中配合使用--flush-logs -F参数，每次全备后都刷新一次bin-log

在全备后新的操作都将被记录到新的bin-log中，故而实现增量备份

过程如下：

先开启 log-bin=mysql-bin

server-id=1 然后重启，



第一次全备：

mkdir /myback

mysql > create database m2; 创建测试库

mysql > create table m2.a (id int(2),name char(10),sex char(1));

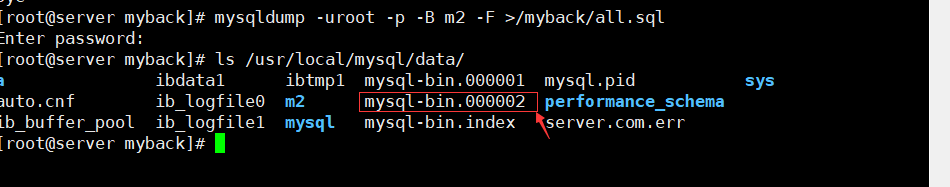
mysql > insert into m2.a values (1,'zhangsan','f'),(2,'lisi','m'); 插入两条数据

mysql > select \* from m2.a;



mysql > exit

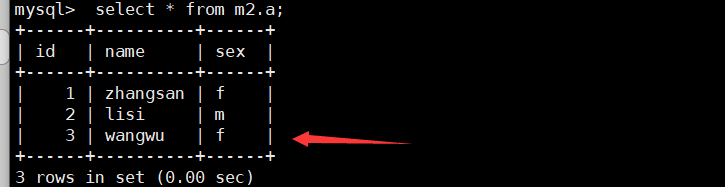
# mysqldump -uroot -p -B m2 -F >/myback/all.sql 第一次全备 然后有新的binlog日志了



# mysql -p

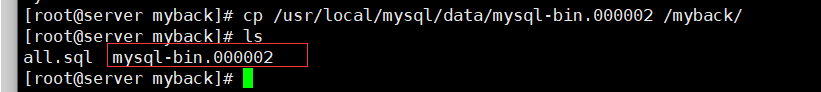
mysql >insert into m2.a values (3,'wangwu','f');

mysql > select \* from m2.a;



mysql > exit

# cp /var/lib/mysql-bin.000002 /myback 将最新的log-bin备份



# mysql -p

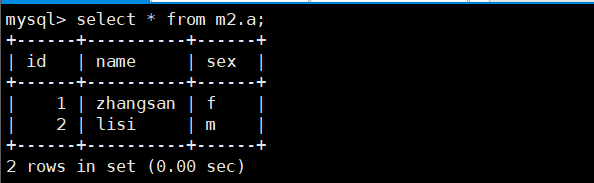
mysql > drop database m2; 删除测试库

mysql > exit

mysql -p </myback/all.sql 恢复全备

# mysql -p

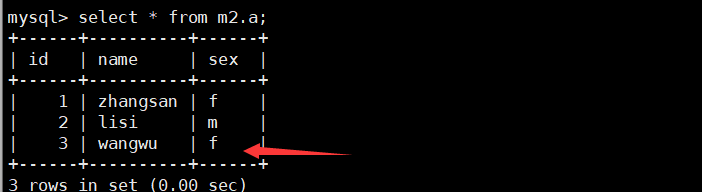
mysql > select \* from m2.a;



# mysqlbinlog /myback/mysql-bin.000002 |mysql -p 恢复增备

# mysql -p

mysql > select \* from m2.a;



也可以直接通过备份bin-log来实现全量和增量备份，通过下列命令可以从指定位置恢复：

1、通过bin-log起始位置恢复

# mysqlbinlog --start-position=4 --stop-position=798 mysql\_bin.000001|mysql -uroot -p

2、通过bin-log中记录时间恢复

# mysqlbinlog --start-datetime='2018-09-20 00:00:00' --stop-datetime='2018-09-31 12:00:00' mysql\_bin.000001|mysql -uroot -p

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

三、mydumper备份工具安装与使用

1、mydumper安装

# yum install glib2-devel mysql-devel zlib-devel pcre-devel openssl-devel gcc gcc-c++ -y 安装依赖

# yum install -y cmake 安装cmake

# wget https://launchpadlibrarian.net/185032423/mydumper-0.6.2.tar.gz 下载源码包

# tar xf ./mydumper-0.6.2.tar.gz -C /usr/src/

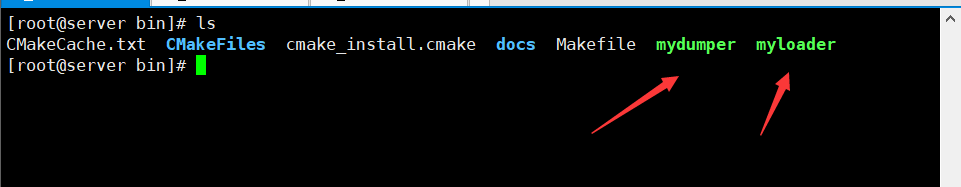
# cd /usr/src/mydumper-0.6.2/

# mkdir ./bin ; cd bin

# cmake ../

# make

# ll mydumper myloader



在当前目录看到这两个文件就表示搞定了，创建软连接或者设置环境变量

ln -s /usr/src/mydumper-0.6.2/bin/mydumper /usr/bin/

ln -s /usr/src/mydumper-0.6.2/bin/myloader /usr/bin/

2、mydumper常用参数说明

-h --host 主机

-u --user 用户

-p --password 密码

-P --port 端口

-S --socket socket文件

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

命令格式：

# mydumper -u root -p -B m2 -o /myback

作用：

将mysql数据库中所有数据备份至/myback目录

注意：该命令-u选项和后面的用户名必须有一个以上空格，否则将会报错

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

-------------------------------------------------------------------------------

-o --outputdir 备份到指定目录

# mydumper -u root -p -B m2 -o /myback

或

# mydumper -u root -p --database=m2 --outputdir=/myback

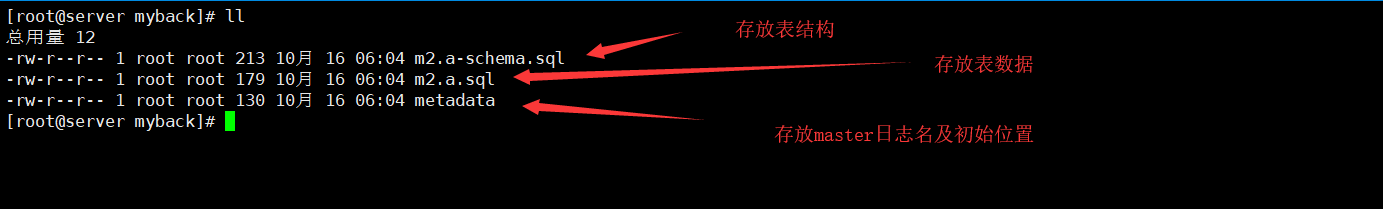
----------------------------------------------------------------------------

-B --database 备份指定数据库，-p 后面输入密码，需要隔空格，

# mydumper -u root -p -B m2 -o /myback

mydumper -u root -p '123456' -B m2 -o /myback/

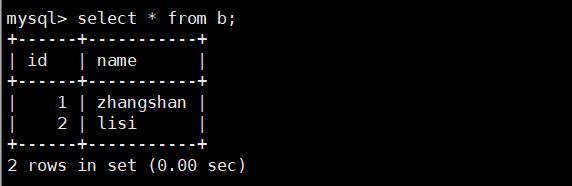
# ll /myback/



创建b表

mysql> create table b (id int, name varchar(20));

mysql> insert into b values (1, 'zhangshan'),(2, 'lisi');



-T --tables-list 备份指定表,多个以“,”分隔

mydumper -u root -p 123456 -B m2 -T b -o /myback/

# ll /myback/

总用量 12

-rw-r--r-- 1 root root 215 9月 24 12:09 m2.b-schema.sql

-rw-r--r-- 1 root root 145 9月 24 12:09 m2.b.sql

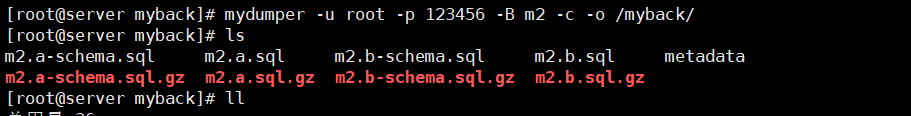
-rw-r--r-- 1 root root 130 9月 24 12:09 metadata

--------------------------------------------------------------------------------

-c --compress 启用压缩备份，备份后的文件为压缩格式

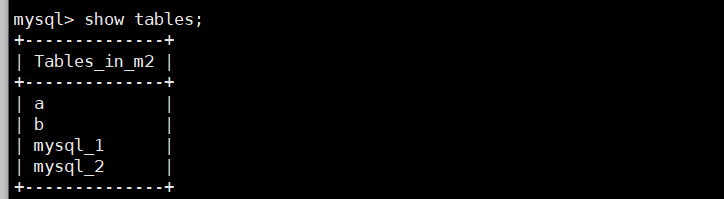
# mydumper -u root -p -B m2 -c -o /myback

# ll /myback

--------------------------------------------------------------------------------

-x, --regex 使用正则匹配，可指定备份相同匹配的表或库（注意：. 和 \* 的不同）

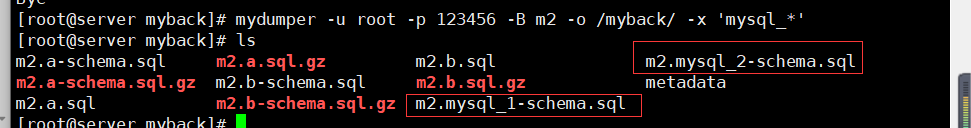
mysql > show tables;



如果只想备份出mysql开头的两个表则需要使用-x参数

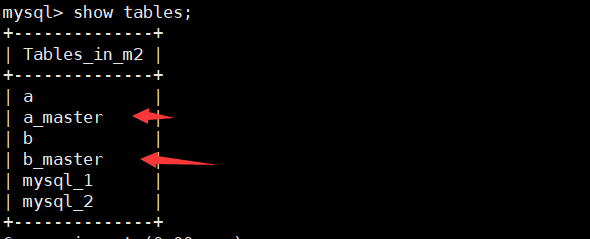
# mydumper -u root -p -B m2 -o /myback -x "mysql\_\*"

# ll /myback



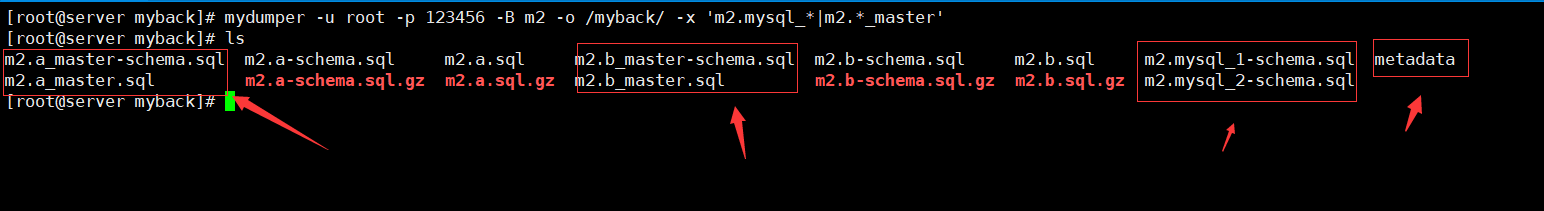
因为表没有插入数据，所以就没有m2.mysql\_1.sql数据文件

mysql > show tables;



如果想同时备份a\_master. b\_master. mysql\_1. mysql\_2该怎么匹配？

# mydumper -u root -p -B m2 -o /myback -x "m2.mysql\_\*|m2.\*\_master"

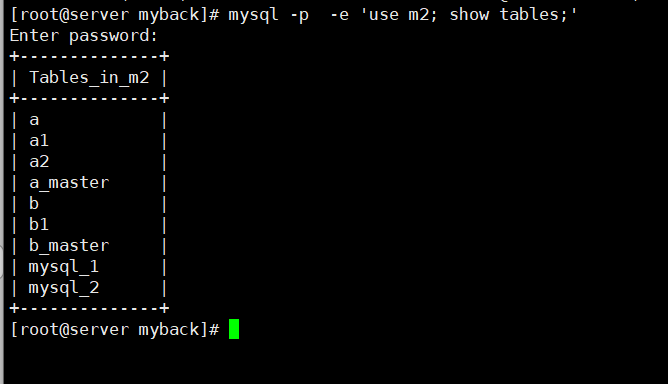


如果想备份下列表中a开头的又该怎么匹配？

mysql -p -e 'use m2; show tables;'

可以使用mysql -e 直接不用进入数据库，

mysql -p -e 'use m2; create table a2 (id int);'



# mydumper -B m2 -o /myback/ -x 'm2.a\*'

-----------------------------------------------------------------------------

-e --build-empty-files 如果备份的表没有数据，还是会备份一个空文件

mysql > select \* from abc;

Empty set (0.00 sec)

# mydumper -u root -p -B m2 -T abc -o /myback

# ll /myback

总用量 8

-rw-r--r-- 1 root root 156 9月 24 12:50 m2.abc-schema.sql 注意，这里少了一个文件

-rw-r--r-- 1 root root 130 9月 24 12:50 metadata

# mydumper -u root -p -B m2 -T abc -e -o /myback

# ll /myback

总用量 8

-rw-r--r-- 1 root root 156 9月 24 12:52 m2.abc-schema.sql

-rw-r--r-- 1 root root 0 9月 24 12:52 m2.abc.sql 这里会备份出来一个空文件

-rw-r--r-- 1 root root 130 9月 24 12:52 metadata

-------------------------------------------------------------------------------

-m --no-schemas 只备份表数据，不备份表结构

# mydumper -B m2 -m -o /myback/

# ll /myback

总用量 28

-rw-r--r-- 1 root root 148 9月 24 12:54 m2.a\_master.sql 备份出来后所有表都没有存放结构的文件

-rw-r--r-- 1 root root 145 9月 24 12:54 m2.a.sql

-rw-r--r-- 1 root root 148 9月 24 12:54 m2.b\_master.sql

-rw-r--r-- 1 root root 141 9月 24 12:54 m2.b.sql

-rw-r--r-- 1 root root 147 9月 24 12:54 m2.mysql\_1.sql

-rw-r--r-- 1 root root 147 9月 24 12:54 m2.mysql\_2.sql

-rw-r--r-- 1 root root 130 9月 24 12:54 metadata

---------------------------------------------------------------------------

-L --logfile 备份出错时，将错误信息追加到指定文件，默认为输出到终端

# mydumper -u root -p -B m2 -o /myback -L /myback/sql\_back.log

# ll /myback

总用量 28

-rw-r--r-- 1 root root 0 9月 24 12:56 sql\_back.log 备份过程中的错误信息将存放到此文件中

-rw-r--r-- 1 root root 148 9月 24 12:56 m2.a\_master.sql

-rw-r--r-- 1 root root 145 9月 24 12:56 m2.a.sql

-rw-r--r-- 1 root root 148 9月 24 12:56 m2.b\_master.sql

-rw-r--r-- 1 root root 141 9月 24 12:56 m2.b.sql

-rw-r--r-- 1 root root 147 9月 24 12:56 m2.mysql\_1.sql

-rw-r--r-- 1 root root 147 9月 24 12:56 m2.mysql\_2.sql

-rw-r--r-- 1 root root 130 9月 24 12:56 metadata

------------------------------------------------------------------------------

-K --kill-long-queries 备份时，如果存在长查询，则将长查询杀掉 大写K

# mydumper -u root -p -B m2 -o /myback -K

备注：如果是工作环境中，难免会有一些长查询的存在，如果存在长查询时，是无法对表进行锁定的，此时备份将可能导致数据的不一致性，所以在备份是，先将这些长查询杀掉，锁表后再进行备份

---------------------------------------------------------------------------

--lock-all-tables 备份前锁定所有表，以防备份后数据不一致，备份完成后自动解锁

# mydumper -u root -p -B m2 -o /myback --lock-all-tables

备注：如果不先对表进行锁定就开始备份，可能会存在备份后数据的不一致性，所以备份时介意先锁表再备份

----------------------------------------------------------------------------

3、mydumper 数据恢复

myloader命令

常用参数说明：

-h --host 指定还原主机

-u --user 指定用户

-p --password 用户密码

-P --port 指定端口

-S --socket 指定socket文件路径

-------------------------------------------------------------------------

-d --directory 指定备份文件存放路径

# myloader -u root -p -d /myback 还原/myback目录下备份所有数据

--------------------------------------------------------------------------

-o --overwrite-tables 恢复时如果表存在，先drop该表（确定要还原的数据中包含表结构）

# myloader -u root -p -o -d /myback 还原时，先检查要还原的表是否存在，如果存在先drop

--------------------------------------------------------------------------

-B --database 指定要还原到哪个数据库

# myloader -u root -p -B m2 -d /myback 还原时指定还原到m2库

---------------------------------------------------------------------------

-e --enable-binlog 还原时，将二进制日志一并还原

# myloader -u root -p -e -d /myback

---------------------------------------------------------------------------

**四、xtrabackup （不常用，专门用于备份innodb表的）& innobackupex（主要用这个） 备份恢复**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

xtrabackup和innobackupex两者间的区别：

xtrabackup可以对innodb表进行热备份（不对表加锁，备份期间数据可读写），但不能对myisam表进行备份；

innobackupex可以同时对上面两者进行备份，但针对myisam表依旧不能进行热备份（必须加锁后方能备份）

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

xtrabackup 安装

# wget https://www.percona.com/downloads/XtraBackup/Percona-XtraBackup-2.4.12/binary/tarball/percona-xtrabackup-2.4.12-Linux-x86\_64.libgcrypt145.tar.gz 下载二进制包（官网下载）

wget <https://www.percona.com/downloads/XtraBackup/Percona-XtraBackup-2.4.4/binary/tarball/percona-xtrabackup-2.4.4-Linux-x86_64.tar.gz> 2.44版本

# yum install libaio libaio-devel -y 安装依赖

yum install -y perl-DBI perl-DBD-MySQL perl-Time-HiRes perl-IO-Socket-SSL perl-TermReadKey.x86\_64 perl-Digest-MD5

# tar xf percona-xtrabackup-2.4.12-Linux-x86\_64.libgcrypt145.tar.gz -C /usr/local/

#mv /usr/local/percona-xtrabackup-2.4.12-Linux-x86\_64/ /usr/local/xtrabackup

# ln -s /usr/local/xtrabackup/bin/xtrabackup /usr/bin/xtrabackup

# ln -s /usr/local/xtrabackup/bin/innobackupex /usr/bin/innobackupex

下载 percona-toolkit 并安装

wget <https://www.percona.com/downloads/percona-toolkit/2.2.19/RPM/percona-toolkit-2.2.19-1.noarch.rpm>

rpm -ivh percona-toolkit-2.2.19-1.noarch.rpm

4.1 Xtrabackup 常用参数

--defaults-file 指定mysql主配置文件路径，备份参数会从配置文件中自动读取（datadir basedir等）

--defaults-extra-file 使用该参数时，不只是读取主配置文件，还会读取这里指定的配置文件

--target-dir 指定备份存放路径，如不指定，则备份到当前目录下的xtrabackup\_backupfiles目录

--backup 备份到--target-dir指定路径（必选），如不选则不备份，只打印过程

--prepare 实施对备份文件进行恢复前的准备(生成InnoDB log file)

--print-param 打印备份或恢复时需要的参数

--suspend-at-end 在备份目录中生成一个xtrabackup\_suspended文件，将xtrabackup进程挂起，不停地将数据文件的变化同步到备份文件，直到用户手工删除xtrabackup\_suspended文件

--tables 备份指定的表--tables=table\_name

--datadir MySQL数据目录路径，一般不用指定

--incremental-basedir 增量备份时使用，自动在指定文件中确定之前一次备份的位置

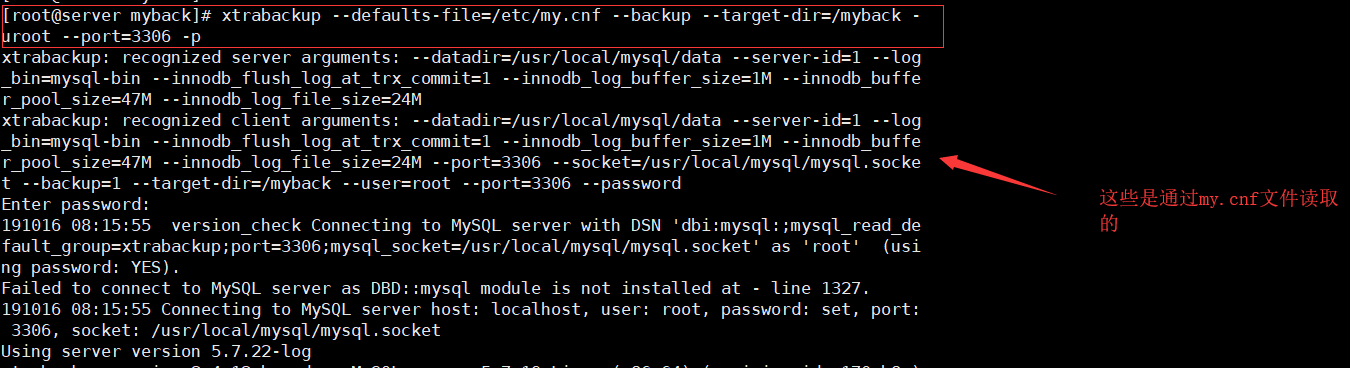
--incremental-lsn 增量备份时，指定to\_lsn位置

xtarbackup备份命令格式：

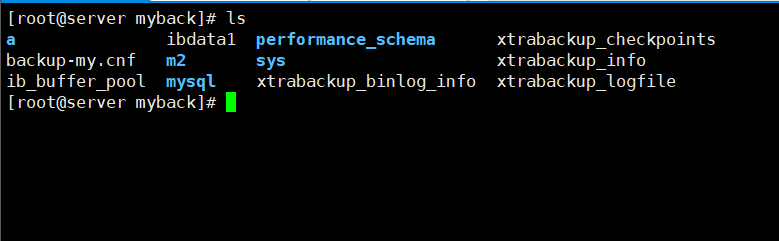
备份：

全备

# xtrabackup --defaults-file=/etc/my.cnf --backup --target-dir=/myback -uroot --port=3306 -S /var/lib/mysql/mysql.sock -p 将数据库全备至/myback目录



# ll /myaback



全备完成

各文件说明： (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。 (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。

1. xtrabackup\_binlog\_pos\_innodb ——二进制日志文件及用于InnoDB或XtraDB表的二进 制日志文件的当前 position。
2. xtrabackup\_binary ——备份中用到的 xtrabackup 的可执行文件；

(5)backup-my.cnf ——备份命令用到的配置选项信息； 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 注意：相关选项说明： 其中， --user 指定连接数据库的用户名， --password 指定连接数据库的密码， --defaults-file 指定数据库的配置文件，innobackupex 要从其中获取 datadir 等信息；--database 指定要 备份的数据库，这里指定的数据库只对 MyISAM 表有效，对于 InnoDB 数据来说都是全 备（所有数据库中的 InnoDB 数据都进行了备份，不是只备份指定的数据库，恢复时也 一样）；/opt/mysqlbackup/full 是备份文件的存放位置。

注意：备份数据库的用户需要具有相应权限，如果要使用一个最小权限的用户进行备份， 则可基于如下命令创建此类用户：

mysql> create user 'bkpuser'@'localhost' identified by '123456';

mysql> revoke all privileges,grant option from 'bkpuser'@'localhost';

mysql> grant reload,lock tables,replication client, process on \*.\* to 'bkpuser'@'localhost';

mysql> flush privileges;

全备恢复

模拟数据丢失环境

停止

# systemctl stop mariadb

删除

# rm /var/lib/mysql/\*

注意：如果是非yum版的mysql删除数据目录后将无法启动，也可以的，

需要重新初始化

源码版 rm /usr/local/mysql/data/\* -rf

然后重新初始化

/usr/local/mysql/bin/mysqld --initialize --user=mysql --basedir=/usr/local/mysql --datadir=/usr/local/mysql/data

重启后将会发现之前的所有数据都已经没了，如果重启报错可能是之前的进程没有关，把kill掉就可以了，

修改初始密码，

mysqladmin -u root password '123456' -p'!g\*ivfjDh1)#'

恢复前准备（初始化备份目录）

# xtrabackup --defaults-file=/etc/my.cnf --prepare --target-dir=/myback

将备份目录中的全部文件拷贝至数据目录中

\cp -rf /myback/\* /var/lib/mysql

\cp -rf /myback/\* /usr/local/mysql/data/

在cp指令前面加反斜杠可以不弹出是否覆盖的询问而直接覆盖！

重新授权 要不然会报错

# chown -R mysql.mysql /var/lib/mysql

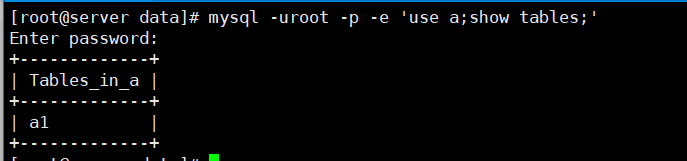
启动mariadb查看，数据恢复成功

----------------------------------------------------------------------------------

增量备份

全备

# mysql -uroot -p -e "use a;show tables;"



# # xtrabackup --defaults-file=/etc/my.cnf --backup --target-dir=/myback -uroot --port=3306 -S /var/lib/mysql/mysql.sock -p

xtrabackup --defaults-file=/etc/my.cnf --backup --target-dir=/myback -uroot -p

# cat /myback/xtrabackup\_checkpoints

backup\_type = full-backuped

from\_lsn = 0 本次备份开始点

to\_lsn = 1632546 本次备份结束点

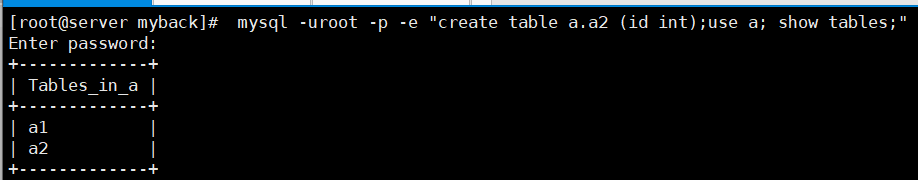
last\_lsn = 1632546 下次备份开始点

compact = 0

recover\_binlog\_info = 0

第一次增量备份，备份至/inc/inc1目录中

# mysql -uroot -p -e "create table a.a2 (id int);use a; show tables;"



mkdir /inc/inc1 -p

# xtrabackup --backup --target-dir=/inc/inc1 --incremental-basedir=/myback -uroot -S /var/lib/mysql/mysql.sock -P 3306

xtrabackup --backup --target-dir=/inc/inc1 --incremental-basedir=/myba

ck -uroot -p #这次备份就不需要指定my.cnf文件了

# cat /inc/inc1/xtrabackup\_checkpoints

backup\_type = incremental

from\_lsn = 1632546

to\_lsn = 1634384

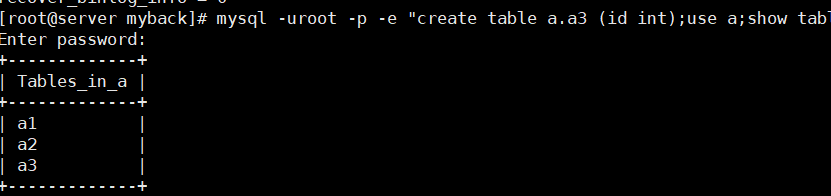
last\_lsn = 1634384

compact = 0

recover\_binlog\_info = 0

第二次增量备份，备份至/inc/inc2目录中

# mysql -uroot -p -e "create table a.a3 (id int);use a;show tables;"



mkdir /inc/inc2

# # xtrabackup --backup --target-dir=/inc/inc2 --incremental-basedir=/inc/inc1 -uroot -S /var/lib/mysql/mysql.sock -P 3306

# cat /inc/inc2/xtrabackup\_checkpoints

backup\_type = incremental

from\_lsn = 1634384

to\_lsn = 1636206

last\_lsn = 1636206

compact = 0

recover\_binlog\_info = 0

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

增量备份时两个参数的说明：

--target-dir 指定增备目录（需要备份到哪儿）

--incremental-basedir 指定上一次备份目录（第一次增量备份时，该参数=全备目录；

第二次增量备份时，该参数=第一次增量备份目录）

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

全量恢复（全备+增备）

模拟数据丢失环境

# systemctl stop mariadb

删除

# rm /var/lib/mysql/\*

源码版安装的就删除对应的库，

开始恢复

初始化全量备份

# xtrabackup --prepare --apply-log-only --target-dir=/myback

初始化第一次增量备份

# xtrabackup --prepare --apply-log-only --target-dir=/myback --incremental-dir=/inc/inc1

初始化第二次增量备份

# xtrabackup --prepare --apply-log-only --target-dir=/myback --incremental-dir=/inc/inc2

拷贝全备目录下所有内容到数据目录中（此时的全备目录已经拥有全量数据）

# cp /myback/\* /var/lib/mysql/ -r

\cp /myback/\* /usr/local/mysql/data/ -r

重新授权

# chown -R mysql.mysql /var/lib/mysql

# systemctl start mariadb

查看表是否正确

# mysql -uroot -p -e "use a;show tables;"

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

| Tables\_in\_a |

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

| a1 |

| a2 |

| a3 |

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

增量恢复完成

----------------------------------------------------------------------------------

4.2 innobackupex 常用参数及使用案例

--apply-log 恢复时，通过xtrabackup\_logfile在备份目录中创建一个备份（恢复备份时必选）

----------------------------------------------------------------------------------

--copy-back 从备份目录中以拷贝的形式恢复到数据目录中

----------------------------------------------------------------------------------

--move-back 从备份目录中以移动的形式恢复到数据目录中

----------------------------------------------------------------------------------

--defaults-file 指定mysql主配置文件路径

----------------------------------------------------------------------------------

--remote-host 使用ssh将备份文件拷贝至远程服务器上

----------------------------------------------------------------------------------

--stream 备份时启用打包

----------------------------------------------------------------------------------

--compress 备份时启用压缩

----------------------------------------------------------------------------------

--tables 指定要备份的表

----------------------------------------------------------------------------------

--databases 指定要备份的库

----------------------------------------------------------------------------------

--uncompress 解压以压缩方式备份的备份文件

----------------------------------------------------------------------------------

--incremental 创建一个增量备份，在增量备份时使用

----------------------------------------------------------------------------------

--incremental-basedir 增量备份时，指定全量备份的路径，以确定全量备份的终止位置

----------------------------------------------------------------------------------

--incremental-dir 将增量备份与全量备份合并后放入新的目录

----------------------------------------------------------------------------------

--incremental-lsn 增量备份时，指定增量备份的起点

----------------------------------------------------------------------------------

--no-lock 备份时，关闭自动锁表功能

----------------------------------------------------------------------------------

--safe-slave-backup 备份时，先停止slave进程

----------------------------------------------------------------------------------

--no-timestamp 备份时，不自动创建以日期和时间的目录

----------------------------------------------------------------------------------

使用案例：

备份最好不要用root用户，需要新建用户

grant select,process,reload,super on \*.\* to 'backup\_mysql'@'localhost' with grant option;

grant select,process,reload,super on \*.\* to 'backup\_mysql'@'localhost' identified by '123321.shui'

flush privileges;

mkdir /mysqlbackup/{wanquan,zengliang} -p

1、全备

开始备份

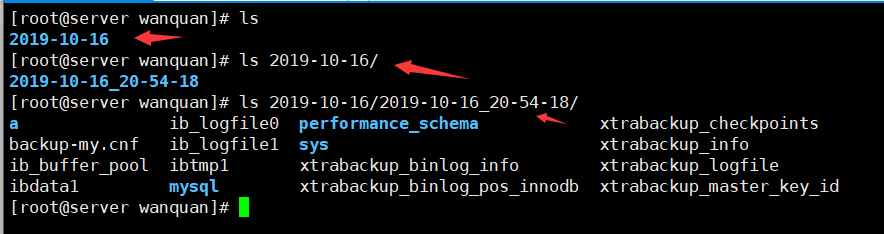
# innobackupex --defaults-file=/etc/my.cnf /myback -uroot -P 3306 -S /var/lib/mysql/mysql.sock 备份所有数据到/myback目录以当前日期命名的目录中

或者：

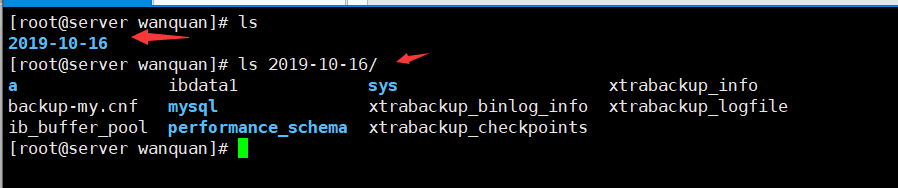
#

innobackupex --defaults-file=/etc/my.cnf --user=backup\_mysql --password=123321.shui /mysqlbackup/wanquan/$(date "+%Y-%m-%d")

$(date "+%Y-%m-%d") #表示已今天日期命名 这样备份会多创建一个目录，所以需要加一个参数 --no-timestamp 告诉innobackupex不要创建一个时间戳目录来存储备份



innobackupex --defaults-file=/etc/my.cnf --user=backup\_mysql --password=123321.shui /mysqlbackup/wanquan/$(date "+%Y-%m-%d") --no-timestamp



测试，删除一个表或库，模拟误操作

# mysql -uroot -p -e "drop database a;"

# mysql -uroot -p -e "show databases;"

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

| Database |

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

| information\_schema |

| mysql |

| performance\_schema |

| test |

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

# systemctl stop mariadb 恢复前记得停止数据库，并删除数据目录中所有文件

# rm -rf /var/lib/mysql/\*

rm /usr/local/mysql/data/\* -rf

innobackupex --defaults-file=/etc/my.cnf --apply-log /mysqlbackup/wanquan/$(ls /mysqlbackup/wanquan/)

恢复前准备（初始化数据）

# innobackupex --defaults-file=/etc/my.cnf --copy-back --datadir=/usr/local/mysql/data /mysqlbackup/wanquan/$(ls /mysqlbackup/wanquan/)

恢复数据到数据目录中；--datadir 指定数据目录位置，如果配置文件中有指定，则不需要在此指定

# chown -R mysql.mysql /var/lib/mysql 给数据目录重新授权

chown -R mysql. /usr/local/mysql/data/

# systemctl start mariadb

# mysql -uroot -p -e "show databases;"

# mysql -uroot -p -e "select \* from a.a1;"

+------+

| id |

+------+

| 1 |

| 2 |

+------+

全备及全备恢复完成！

已经初始化的全备目录的数据不能再做为增量备份的基础数据，下面需要增量备份，所以这里需要重新备份，

innobackupex --defaults-file=/etc/my.cnf --user=backup\_mysql --password=123321.shui /mysqlbackup/wanquan/$(date "+%Y-%m-%d") --no-timestamp

----------------------------------------------------------------------------------

2、innobackupex 增量备份

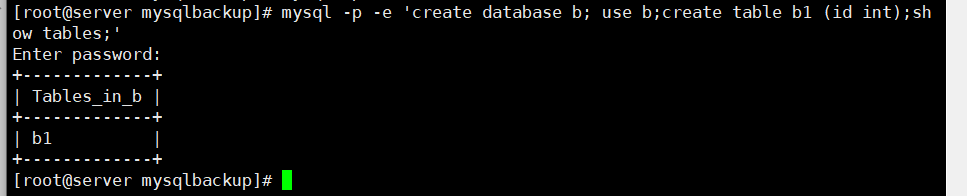
--incremental-basedir 指定全量备份目录，因为增量备份需要知道前一次全量是到哪里结束的

# cat /mysqlbackup/wanquan/`ls /mysqlbackup/wanquan/`/xtrabackup\_checkpoints



--incremental 指定增量备份的目录，会在指定目录下创建一个以当前日期和时间命名的

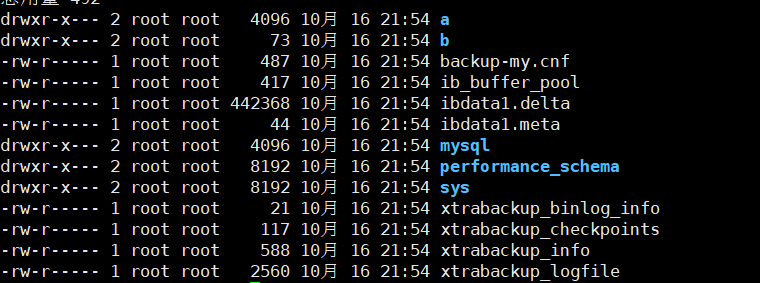
mysql -p -e 'create database b; use b;create table b1 (id int);show tables;'



第一次增量备份

# innobackupex --incremental /mysqlbackup/zengliang/$(date "+%Y-%m-%d") --incremental-basedir=/mysqlbackup/wanquan/`ls /mysqlbackup/wanquan/` --user=backup\_mysql --password=123321.shui --no-timestamp

ll /mysqlbackup/zengliang/$(date "+%Y-%m-%d")



#cat /mysqlbackup/zengliang/$(date "+%Y-%m-%d")/xtrabackup\_checkpoints

backup\_type = incremental

from\_lsn = 2627131 本次增量备份的起点

to\_lsn = 2630943 本次增量备份的终点

last\_lsn = 2630952 下一次增量备份的起点

compact = 0

recover\_binlog\_info = 0

创建C 库

mysql -p -e 'create database c; use c;create table c1 (id int);show tables;'

第二次增量备份 目录日期不一样，还--incremental-basedir= 指定的目录是第一次增量备份的目录

# innobackupex --incremental /mysqlbackup/zengliang/$(date "+%Y-%m-%d-%H") --incremental-basedir=/mysqlbackup/zengliang/`ls /mysqlbackup/zengliang/` --user=backup\_mysql --password=123321.shui --no-timestamp

cat /mysqlbackup/zengliang/$(date "+%Y-%m-%d-%H")/xtrabackup\_checkpoints

backup\_type = incremental

from\_lsn = 2630943 上次增量备份的终点，本次的起点

to\_lsn = 2634771 本次的终点，下次的起点

last\_lsn = 2634780

compact = 0

recover\_binlog\_info = 0

增量备份完成！

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

注释信息

备份目录详解：

/mysqlbackup/wanquan/2019-10-16 全备目录（--incremental-basedir使用的）

/mysqlbackup/zengliang/2019-10-16 第一次增量备份目录(--incremental使用的)

/mysqlbackup/zengliang/2019-10-17-23 第二次增量备份目录(--incremental使用的)

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

----------------------------------------------------------------------------------

增量备份恢复

模拟数据丢失环境

停止

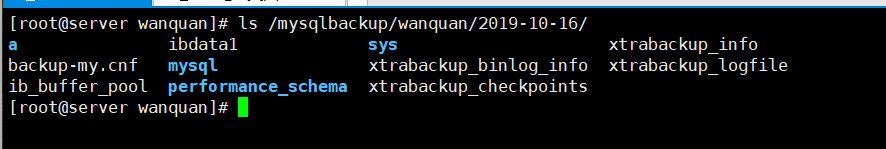
# systemctl stop mariadb

删除

# rm /var/lib/mysql/\*

mkdir -p /tmp/mysqldatabak

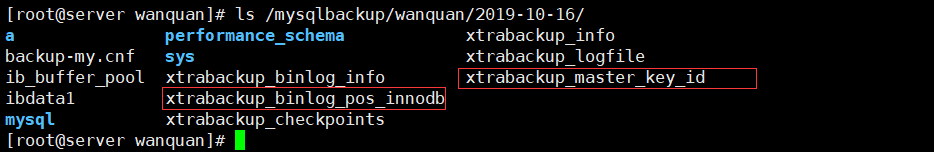
mv /usr/local/mysql/data/\* /tmp/mysqldatabak



初始化全备目录

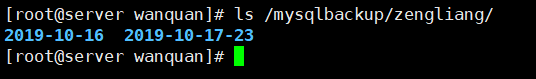
innobackupex --apply-log --redo-only /mysqlbackup/wanquan/`ls /mysqlbackup/wanquan/`

初始化后会多两个个文件



查看有多少增量备份目录

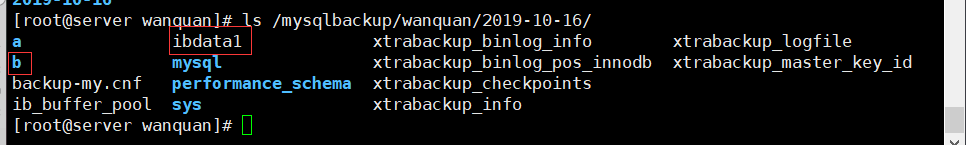
ls /mysqlbackup/zengliang/



初始化第一次增备目录

# innobackupex --apply-log --redo-only /mysqlbackup/wanquan/`ls /mysqlbackup/wanquan/` --incremental-dir=/mysqlbackup/zengliang/2019-10-16

把增量备份的数据整合到全备了



初始化第二次增备目录（注意：这一步不需要加 --redo-only 选项）

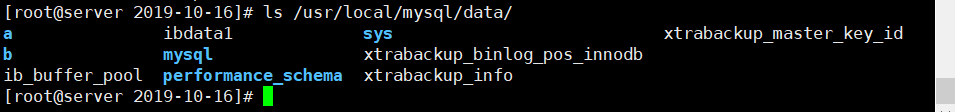
#innobackupex --apply-log /mysqlbackup/wanquan/`ls /mysqlbackup/wanquan/` --incremental-dir=/mysqlbackup/zengliang/2019-10-17-23

开始完整恢复

# innobackupex --copy-back /mysqlbackup/wanquan/`ls /mysqlbackup/wanquan/`

过程：先将两次增量备份所备份的数据整合到全备目录中，然后进行一次性恢复

ls /usr/local/mysql/data/



重新授权

# chown -R mysql.mysql /var/lib/mysql

chown -R mysql. /usr/local/mysql/data/

启动

# systemctl start mariadb

# mysql -uroot -p -e "show databases;"

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

| Database |

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

| information\_schema |

| a |

| mysql |

| performance\_schema |

| test |

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

# mysql -uroot -p -e "use a;show tables;"

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

| Tables\_in\_a |

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

| a1 |

| a2 |

| a3 |

| a4 |

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

至此，innobackupex 增量备份与恢复完成！

工作环境一周的数据备份及恢复

备份最好不要用root用户，需要新建用户

grant select,process,reload,super on \*.\* to 'backup\_mysql'@'localhost' with grant option;

grant select,process,reload,super on \*.\* to 'backup\_mysql'@'localhost' identified by '123321.shui'

flush privileges;

mkdir /mysqlbackup/{wanquan,zengliang} -p

1. 全备 每周日

innobackupex --defaults-file=/etc/my.cnf --user=backup\_mysql --password=123321.shui /mysqlbackup/wanquan/$(date "+%Y-%m-%d") --no-timestamp

2、innobackupex 增量备份 周一到周六， date +%w

--incremental-basedir 指定全量备份目录，因为增量备份需要知道前一次全量是到哪里结束的

--incremental 指定增量备份的目录，会在指定目录下创建一个以当前日期和时间命名的

创建库表

mysql -p -e 'create database b;use b;create table b1 (id int);show tables;'

周一增量备份，增量备份命名为周一 （1）

innobackupex --incremental /mysqlbackup/zengliang/$(date "+%w") --incremental-basedir=/mysqlbackup/wanquan/`ls /mysqlbackup/wanquan/` --user=backup\_mysql --password=123321.shui --no-timestamp

再次创建表，

mysql -p -e 'create database c; use c;create table c1 (id int);show tables;'

周二增量备份 这时候--incremental-basedir 指定的目录就是周一的，也就是前一天的，

date "+%w" -d '1 day ago' 代表前一天

innobackupex --incremental /mysqlbackup/zengliang/$(date "+%w") --incremental-basedir=/mysqlbackup/zengliang/$(date "+%w" -d '1 day ago' --user=backup\_mysql --password=123321.shui --no-timestamp

由于是测试我就把命名时间改下，改为6天前，就是周二了，

innobackupex --incremental /mysqlbackup/zengliang/$(date "+%w" -d '6 day ago') --incremental-basedir=/mysqlbackup/zengliang/$(date "+%w") --user=backup\_mysql --password=123321.shui --no-timestamp

周三备份，和周二的是一样的，只是测试环境就是下面的

创建库表

mysql -p -e 'create database d;use d; create table d1 (id int);show tables;'

innobackupex --incremental /mysqlbackup/zengliang/$(date "+%w" -d '5 day ago') --incremental-basedir=/mysqlbackup/zengliang/$(date "+%w" -d '6 day ago') --user=backup\_mysql --password=123321.shui --no-timestamp

后面周四周五周六都是和周二是一样的，

现在开始模拟数据丢失，进行恢复

systemctl stop mysqld

mkdir -p /tmp/mysqldatabak

mv /usr/local/mysql/data/\* /tmp/mysqldatabak

初始化全备目录

innobackupex --apply-log --redo-only /mysqlbackup/wanquan/`ls /mysqlbackup/wanquan/`

初始化周一增备目录,

ls /mysqlbackup/zengliang

# innobackupex --apply-log --redo-only /mysqlbackup/wanquan/`ls /mysqlbackup/wanquan/` --incremental-dir=/mysqlbackup/zengliang/1

初始化周二次增备目录

ls /mysqlbackup/zengliang

# innobackupex --apply-log --redo-only /mysqlbackup/wanquan/`ls /mysqlbackup/wanquan/` --incremental-dir=/mysqlbackup/zengliang/2

初始化周三次增备目录（注意：最后一步可以不加 --redo-only 选项）

innobackupex --apply-log --redo-only /mysqlbackup/wanquan/`ls /mysqlbackup/wanquan/` --incremental-dir=/mysqlbackup/zengliang/3

开始完整恢复

# innobackupex --defaults-file=/etc/my.cnf --copy-back /mysqlbackup/wanquan/`ls /mysqlbackup/wanquan/`

过程：先将三次增量备份所备份的数据整合到全备目录中，然后进行一次性恢复

授权

chown -R mysql. /usr/local/mysql/data/

启动

# systemctl start mariadb

# mysql -uroot -p -e "show databases;"

可以参考脚本

mysqlbackup.sh

先修改/etc/my.cnf 文件，打开log-bin=mysql-bin

server-id=1

重启

登录，授权，

grant select,process,reload,super on \*.\* to 'backup\_mysql'@'localhost' identified by '123321.shui' ;

flush privileges;

crontab -e

0 8 \* \* \* /usr/bin/sh /root/mysqlbackup.sh

每天8点钟执行这个脚本做备份