一 mysql备份

1，数据备份

－－物理备份

冷备：cp，tar......

[root@mysql-50 ~]# systemctl stop mysqld

[root@mysql-50 ~]# cp -r /var/lib/mysql /root/mysql.bak

[root@mysql-50 ~]# scp -r /root/mysql.bak 192.168.4.51:/

[root@mysql-51 ~]# mv /mysql.bak /var/lib/mysql

[root@mysql-51 ~]# chown -R mysql:mysql /var/lib/mysql

[root@mysql-51 ~]# systemctl restart mysqld

特点：

mysql版本要一致

操作系统要一样

不适合innodb存储引擎的数据库备份

－－逻辑备份

Mysqldump

Mysql

－－数据备份策略

完全备份：备份所有数据

增量备份：备份上次备份后，备份所有新产生的数据

差异备份：备份完全备份后，备份所有新产生的数据

２，mysqldump备份与恢复

－－备份,恢复操作

Mysqldump -uroot -p密码 库名 > 路径/xxx.sql

Mysql -uroot -p密码 库名 < 路径/xxx.sql

库名表示方式

--all-databases | -A ：所有库

数据库名　　　　　　 :单个表

数据库名　表名　　　 :单张表

-B 数据库１ 数据库２.. :多个数据库

注意事项

无论备份还是恢复，都要验证用户权限

例子：

##恢复数据表

单个数据库

[root@mysql-50 ~]# mysqldump -uroot -p123456 db2 >/data/db2.sql

[root@mysql-50 ~]# mysql -uroot -p123456 db2 < /data/db2.sql

提示：恢复单个整个数据库要先新建数据库，需要先建数据库

多个数据库

[root@mysql-50 ~]# mysqldump -uroot -p123456 -B db3 db4 >/data/db3-db4.sql

[root@mysql-50 ~]# mysql -uroot -p123456 < /data/db3-db4.sql

##恢复全部数据库

[root@mysql-50 ~]# mysqldump -uroot -p123456 -A > /data/alldb.sql

[root@mysql-50 ~]# mysql -uroot -p123456 < /data/alldb.sql

缺点：

锁表

不能实现实时备份

３，实时增量备份

－－binlog日志

采用binlog日志的好处：

记录除查询之外的所有sql命令

可用于数据恢复

配置mysql主从同步的必要条件

１）启用日志文件

[root@mysql-50 ~]# mkdir /mysqllog

[root@mysql-50 ~]# chown -R mysql:mysql /mysqllog

[root@mysql-50 ~]# vim /etc/my.cnf

[mysqld]

#log\_bin　　#开启日志，使用默认，默认位置mysql的初始化目录，名称为主机名称

#log\_bin=jluo　　　　　　　　#开启日志，并指定日志名称

log\_bin=/mysqllog/mysql　　　#开启日志，并指定日志位置和名称

server\_id=50 #mysql5.7必须写，且id号不能重复

#Max\_binlog\_size=数字m #修改日志大小，(默认1Ｇ),满后自动新建另一个

binlog\_format="mixed"

#statement:每一条修改数据的sql命令都记录在binlog日志中

#row:不记录sql语句上下文相关信息，仅保存哪条记录被修改

#mixed：是以上两种格式的混合使用

[root@mysql-50 ~]# systemctl restart mysqld

[root@mysql-50 ~]# ls /mysqllog/

mysql.000001 mysql.index　　　#第一个文件是日志文件，第二个是日志索引文件

1. mysqlbinlog工具

Mysqlbinlogslow:慢日志查询

格式：

Mysqlbinlog　 [选项]　　binlog日志文件名

常用选项

--start-datetime=”yyyy-mm-dd hh:mm:ss”　　#开始于时间点

--stop-datetime=”yyyy-mm-dd hh:mm:ss”　　#结束于时间点

--start-position=数字　　　　　　　　　#开始位置（偏移量）

--stop-position=数字　　　　　　　　　#结束偏移量

例子：

mysql> create table t1 (id int);

mysql> insert into t2 values (1);

mysql> insert into t2 values (2);

mysql> insert into t2 values (3);

[root@mysql-50 mysqllog]# mysqlbinlog mysql.000001 | grep create

#181121 11:31:34 server id 50 end\_log\_pos 123 CRC32 0x9834a147 Start: binlog v 4, server v 5.7.17-log created 181121 11:31:34 at startup

create table t1 (id int)

[root@mysql-50 mysqllog]# mysqlbinlog mysql.000001 | grep insert

insert into t1 values(1)

insert into t1 values(2)

insert into t1 values(3)

3）手动生成新的日志文件

重启mysql服务

执行sql操作　mysql>flush logs;

Mysqldump --flush-logs

Mysql -uroot -p密码 -e ‘flush logs’ #-e 后面不能跟太复杂的命令

例子：  
mysqldump --flush-logs -uroot -p123456 db4 > /data/db4.sql

４）清理Binlog日志

mysql> show master status;　　　#查看当前使用的日志文件

删除早于指定版本的binlog日志

Purge master logs to “binlog文件名”;

删除所有binlog日志，重建新日志

　　Reset master;

提示：

　　不介意使用系统命令删除，使用mysql命令删除

例子：

mysql> purge master logs to "mysql.000004";　#删除指定文件之前的，不包括此文件

mysql> reset master;　　#重建新日志

５）binlog恢复数据

思路：

使用mysqlbinlog提取历史sql操作

通过管道交给mysql命令执行

例子：

[root@mysql-50 mysqllog]# mysqldump -uroot -p123456 db2 < /data/db2.sql

mysql> insert into t2 values(11,'陈红',12);

mysql> update t2 set name='貂蝉' where id = 2;

mysql> drop database db2;　　　　#

mysql> create database db2;

[root@mysql-50 mysqllog]# mysql -u root -p123456 db2 < /data/db2.sql

[root@mysql-50 mysqllog]# mysqlbinlog --start-position=296 --stop-position=1857 mysql.000001 | mysql -u root -p123456

－－innobackupex

物理备份缺点：

跨平台差

备份时间长，冗余备份，浪费存储空间

Mysqldump备份缺点

效率较低，备份和还原速度慢

备份过程中，数据插入和更新操作会被挂起（锁表）

XtraBackup工具

一款强大的在线热备份工具

备份过程中不锁库表，适合生产环境，（备份到哪一行就锁哪一行）

由专业组织percona提供（改进mysql分支）

主要含有两个组件

Xtrabackup:c程序,支持innodb/xtradb

Innobackupex:以perl脚本封装xtrabackup,还支持myisam

提示：实际只有innodb存储引擎支持实时增量备份

1）安装软件包

[root@mysql-50 jluo]# yum -y install libev-4.15-1.el6.rf.x86\_64.rpm

[root@mysql-50 jluo]# yum -y install percona-xtrabackup-24-2.4.7-1.el7.x86\_64.rpm

[root@mysql-50 jluo]# rpm -ql percona-xtrabackup-24

/usr/bin/innobackupex　　　#备份innodb,xtrdb,myisamm引擎的表

/usr/bin/xbcloud

/usr/bin/xbcloud\_osenv

/usr/bin/xbcrypt

/usr/bin/xbstream

/usr/bin/xtrabackup　　#备份innodb,xtrdb引擎的表

/usr/share/doc/percona-xtrabackup-24-2.4.7

/usr/share/doc/percona-xtrabackup-24-2.4.7/COPYING

/usr/share/man/man1/innobackupex.1.gz

/usr/share/man/man1/xbcrypt.1.gz

/usr/share/man/man1/xbstream.1.gz

/usr/share/man/man1/xtrabackup.1.gz

2）完全备份与恢复

　恢复数据要求数据库目录必须为空

　备份目录必须为空

格式：

Innobackupex　选项　　路径

基本选项：

--host：主机名，不写默认本机地址

--user：用户名

--port：端口号，不写默认3306

--password：用户密码

--databases：数据库名　#=”库名”:单个库　=”库1 库2”:多个库　=”库.表”:单个表 不写默认备份所有库

--no-timestamp：不用日期命令备份文件存储的子目录名

--redo-only：日志合并

--apply-log：准备还原（回滚日志）

--copy-back：恢复数据

--incremental 目录名：指定增量备份目录（目录必须为空）

--incremental-basedir=目录名：增量备份时，指定上一次备份数据存储的目录名

--incremental-dir=目录名：准备恢复数据时，指定增量备份数据存储的目录名

--export：导出表信息

import：导出表空间

例子：

##备份

[root@mysql-50 jluo]# innobackupex --user root --password 123456 /allbak

[root@mysql-50 jluo]# ls /allbak/

2018-11-21\_17-48-58

##恢复　　　　　　　　　　提示：数据库目录必须为空，否则报错

[root@mysql-50 jluo]# innobackupex --user root --password 123456 /allbak/2018-11-21\_17-48-58/ --apply-log

[root@mysql-50 jluo]# innobackupex --user root --password 123456 /allbak/2018-11-21\_17-48-58/ --copy-back

1. 增量备份与恢复

##备份全部数据库与恢复

mysql> show databases;　　　　#备份前查看数据库信息

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

| Database |

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

| information\_schema |

| db5 |

| mysql |

| performance\_schema |

| stum |

| stusystem |

| sys |

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

7 rows in set (0.01 sec)

备份：

首先是完全备份

[root@mysql-50 ~]# innobackupex --user root --password 123456 --no-timestamp /bak/alldir

向表里写数据增量备份

mysql> insert into db5.t1 values(1212);

mysql> insert into db5.t1 values(1212);

[root@mysql-50 ~]# innobackupex --user root --password 123456 --incremental /bak/new1dir --incremental-basedir=/bak/alldir --no-timestamp

向表里写数据增量备份

mysql> insert into db5.t1 values(1313);

mysql> insert into db5.t1 values(1313);

[root@mysql-50 ~]# innobackupex --user root --password 123456 --incremental /bak/new2dir --incremental-basedir=/bak/new1dir --no-timestamp

[root@mysql-50 ~]# ls /bak/alldir/

backup-my.cnf mysql sys xtrabackup\_logfile

db5 performance\_schema xtrabackup\_binlog\_info

ib\_buffer\_pool stum xtrabackup\_checkpoints

ibdata1 stusystem xtrabackup\_info

[root@mysql-50 ~]# vim /bak/alldir/xtrabackup\_checkpoints 　#备份信息文件

backup\_type = full-backuped #完全备份

from\_lsn = 0 #lsn：日志序列号

to\_lsn = 7002759

last\_lsn = 7002768

compact = 0

recover\_binlog\_info = 0

[root@mysql-50 ~]# vim /bak//new1dir/xtrabackup\_checkpoints

backup\_type = incremental

from\_lsn = 7002759

to\_lsn = 7005238

last\_lsn = 7005247

compact = 0

recover\_binlog\_info = 0

[root@mysql-50 ~]# vim /bak/new2dir/xtrabackup\_checkpoints

backup\_type = incremental

from\_lsn = 7005238

to\_lsn = 7005861

last\_lsn = 7005870

compact = 0

recover\_binlog\_info = 0

恢复步骤：

１，准备恢复数据

２，合并日志文件（同时也拷贝数据到主目录里）

３，把备份目录下数据拷贝到数据库目录下

４，修改数据库目录的所有者和组用户为mysql

５，启动数据库服务

例子：

[root@mysql-50 ~]# systemctl stop mysqld　#停掉mysql

[root@mysql-50 ~]# rm -rf /var/lib/mysql　#删掉数据库，模拟数据库损坏，要重新恢复数据

[root@mysql-50 ~]# mkdir /var/lib/mysql　#新建数据库目录

[root@mysql-50 ~]# innobackupex --apply-log --redo-only /bak/alldir/

[root@mysql-50 ~]# innobackupex --apply-log --redo-only --incremental-dir=/bak/new1dir/ /bak/alldir/

[root@mysql-50 ~]# innobackupex --apply-log --redo-only --incremental-dir=/bak/new２dir/ /bak/alldir/

[root@mysql-50 ~]# innobackupex --copy-back /bak/alldir/

[root@mysql-50 ~]# chown -R mysql:mysql /var/lib/mysql

[root@mysql-50 ~]# systemctl restart mysqld

[root@mysql-50 ~]# mysql -uroot -p123456

mysql> show databases;

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

| Database |

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

| information\_schema |

| db5 |

| mysql |

| performance\_schema |

| stum |

| stusystem |

| sys |

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

7 rows in set (0.01 sec)

４）恢复完全备份中的单个表

步骤：

１，备份单个数据库

２，删除数据库的单个表

３，新建一张表，并且与刚才删除表的结构一样，并删除该表的空间文件

３，导出表信息

４，拷贝表信息文件到mysql目录对应数据库下

５，修改所有者权限

６，导入表空间

[root@mysql-50 ~]# innobackupex --user root --password 123456 --databases="db5" /db5bak --no-timestamp　#备份数据库

mysql> drop table db5.t2;　　#模拟删除库

mysql> create table t2( name char(10));#新建的t2必须与删除的t2表结构一样

[root@mysql-50 ~]# ls /var/lib/mysql/db5/　#查看新建对应数据库文件

db.opt t1.frm t1.ibd t2.frm t2.ibd

mysql> alter table db5.t2 discard tablespace;　#删除表空间文件，即删除t2.idb文件

[root@mysql-50 ~]# ls /var/lib/mysql/db5/　#查看删除表空间后的数据库文件

db.opt t1.frm t1.ibd t2.frm

[root@mysql-50 ~]# innobackupex --user root --password 123456 --apply-log --databases=”db5” --export /db5bak/　　#导出表信息

[root@mysql-50 ~]# cp /db5bak/db5/t2.{ibd,cfg,exp} /var/lib/mysql/db5/ #拷贝文件

[root@mysql-50 ~]# chown -R mysql:mysql /var/lib/mysql/db5/t2.\*　#授权

mysql> alter table db5.t2 import tablespace;　　#导入表空间

mysql> select \* from db5.t2;　　　#查看数据库