mysql数据备份

**逻辑备份：**

执行备份时，根据已有的库表记录，生成对应的sql命令，sql命令保存到指定的文件，当数据丢失时，可以执行保存sql命令的文件，把数据重新写入数据库里

备份方式

**完全备份** –备份所有数据 mysqldump

（一张表，一个库，一台服务器）

备份时，备份新产生的数据

**差异备份**  -备份自**完全备份**后，都有新产生的数据

**增量备份**  - 备份自**上次备份**后，所有新产生的数据

备份策略： 完全+差异 完全+增量

**crond + 备份脚本**

00 18 \* \*　１　 ／root/allbak.sh(完全备份)

00 18 \* \*　2—7　／root/new.sh(增量备份)

**数据库完全备份**

备份操作

**– mysqldump 源库名 > 路径/xxx.sql**

**源库名的表示**

– --all-databases 或 -A 所有库

– 数据库名 单个库

– 数据库名 表名 单张表

– -B 数据库1 数据库2 备份多个库

• 注意事项

– **无论备份还是恢复，都要验证用户及权限**

mysqldump -uroot -p123456 -A > databa.sql **备份全部**

mysqldump -uroot -p123456 db1 > databa1.sql **备份单个库**

mysqldump -uroot -p123456 db4 t4 > databa2.sql **备份单个库的单张表**

**数据库完全恢复**

恢复操作

– mysql 目标库名 < 路径/xxx.sql

1，mysql -uroot -p123456 userdb < /databak/databa.sql

**增量备份**

启动mysql服务的binlog日志文件实现时时增量备份

**binlog日志(2进制文件)介绍：mysql服务日志文件的一种，记录用户连接后执行的除查询之外的sql命令。默认没有开启**

**修改配置文件**

server\_id=50  **# 指定编号**

log\_bin=/logdir/plj  **#启用binlog日志并指定存放路径及名字**

**要记得修改所属主组**

binlog\_format=”mixed”  **#指定记录格式**

**三种记录格式：**

**1. statement：每一条修改数据的sql命 令都会记录在binlog日志中。**

**2. row:不记录sql语句上下文相关信息， 仅保存哪条记录被修改。**

**3. mixed: 是以上两种格式的混合使用。**

**max\_binlog\_size= 数字m # 指定多少M更换且新增加一个日志文件**

**查看日志文件的内容**

**msyqlbinlog hsot50.bin.000001**

**binlog相关文件**

**– 主机名-bin.index 记录已有日志文件名**

**– 主机名-bin.000001 第1个二进制日志**

**– 主机名-bin.000002 第2份二进制日志**

**数据库输入sql语句后会自动记录在指定的日志存放目录里（除了查询不记录）**

**binlog日志文件记录sql命令的方式**

**偏移量 pos**

**时间点 time**

**执行binlog日志文件记录的sql命令恢复数据**

**mysqlbinlog 选项 binlog日志文件 | mysql -uroot -p密码**

**使用 mysqlbinlog 工具**

**– 格式：mysqlbinlog [选项] binlog日志文件名**

**• 常用选项**

– --start-datetime="yyyy-mm-dd hh:mm:ss"  **#起始时间点**

– --stop-datetime="yyyy-mm-dd hh:mm:ss"  **#结束时间点(不指默认到结尾)**

– --start-position=数字  **#起始偏移量**

–--stop-position=数字  **#结束偏移量(不指默认到结尾)**

**例子：mysqlbinlog --start-position=298 --stop-position=400 plj.000001 | mysql -uroot -p123456**

**手动生成新的日志文件：**

**1. 重启mysql服务**

**2. 执行SQL操作 mysql> flush logs;**

**3. mysqldump --flush-logs**

**4. mysql -uroot -p密码 -e 'flush logs'**

**#！/bin/bash**

**day=`date +%F`**

**if [ ! -e /databak];then**

**mkdir /databak**

**fi**

**mysqldump -uroot -p123456 –flush-logs db1 > *databak*/db1-${day}.sql**

**清理binlog日志**

**• 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;**

**系统命令rm -rf 删除不会同步索引文件**

**show master status; ##记录日志名和偏移量的位置**

**xtrabackup**

**percona-xtrabackup-24-2.4.7-1.el7.x86\_64.rpm ##安装包**

**libev-4.15-1.el6.rf.x86\_64.rpm ##依赖包**

**innobackupex基本选项**

**常用选项 含 义**

**--host 主机名**

**--user 用户名**

**--port 端口号**

**--password 密 码**

**--databases 数据库名**

**--no-timestamp 不用日期命名备份文件存储 的子目录名**

**--databases="库名" //单个库**

**--databases="库1 库2" //多个库**

**--databases="库.表" //单个表**

**innobackupex基本选项（续1）**

**常用选项 含 义**

**--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 增量备份**

**--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 导入表空间**

**完全备份：**

innobackupex --user root --password 123456 /backup --no-timestamp

**完全恢复**

**1恢复日志**

innobackupex --user root --password 123456 --apply-log /backup

**2恢复数据**

innobackupex --user root --password 123456 --copy-back /backup

**3重启服务**

**增量备份**

**第一次增量备份**

innobackupex --user root --password 123456 --databases="performance\_schema mysql sys db5" --incremental /new1 --incremental-basedir="/backup" --no-timestamp

**第二次增量备份**

innobackupex --user root --password 123456 --databases="performance\_schema mysql sys db5" --incremental /new2 --incremental-basedir="/new1" --no-timestamp

**增量恢复**

**1准备恢复日志**

innobackupex --user root --password 123456 --apply-log --redo-only /backup

**2合并数据**

innobackupex --user root --password 123456 --apply-log --redo-only /backup –incremental-dir=/new1

innobackupex --user root --password 123456 --apply-log --redo-only /backup --incremental-dir=/new2

**3恢复数据**

innobackupex --user root --password 123456 --copy-back /backup

**4重启服务**

**innobackupex --user root --password 123456 --databases="db55" /db55dir –no-timestamp**

**在此要删除表 drop table q1；**

**在完全备份文件中恢复单个表**

**1在备份文件里导出表信息**

innobackupex --user root --password 123456 --databases="db55" --apply-log – export / /db55dir –no-timestamp

**2创建删除的表，并删除表空间文件**

alter table db55.q1 **discard tablespace;**

**3 把备份目录下，导出的表信息和表空间文件，拷贝到对应的数据库目录下**

cp *db55dir*/db55/q1.{cfg,exp,ibd}  *var/lib/mysql/db55/*

**在导入表空间前要修改拷贝到库的所属主和组 chown**

**4 导入表空间 import**

alter table db55.q1 import tablespace;

Maxscale 读写分离 主从架构中，主库过于繁忙，从库过于空闲。为合理分配数据库服务器资源，将从库利用起来，响应客户端读请求。

Mycat 分片服务器；读写分离。