# mysqldump备份/恢复

### mysqldump 备份 / 恢复

### 数据库备份概述

数据备份方式

• 物理备份50

– 冷备: cp 、 tar 、 .. ..

• 逻辑备份

– mysqldump

– mysql

数据备份策略

• 完全备份mysqldump

– 备份所有数据(一台服务器、一个库、一张表)

• 增量备份

– 备份自上一次备份(包含完全备份、差异备份、增量

备份)之后有变化的数据

• 差异备份

– 备份自上一次完全备份之后有变化的数据

### 物理备份与恢复

• 备份操作52

– cp -rp /var/lib/mysql/ 数据库 备份目录 / 文件名

– tar -zcvf xxx.tar.gz /var/lib/mysql/ 数据库 /\*

[root@mysql52 ~]# mkdir /datadir

[root@mysql52 ~]# cp -r /var/lib/mysql /datadir/mysql.bak

[root@mysql52 ~]# scp -r /datadir/mysql.bak/ 192.168.4.51:/root/

• 恢复操作51

– cp -rp 备份目录 / 文件名 /var/lib/mysql/

– tar -zxvf xxx.tar.gz -C /var/lib/mysql/数据库名/

[root@mysql51 ~]# rm -rf /var/lib/mysql

[root@mysql51 ~]# systemctl stop mysqld

[root@mysql51 ~]# cp -r /root/mysql.bak/ /var/lib/mysql

[root@mysql51 ~]# chown -R mysql.mysql /var/lib/mysql

[root@mysql51 ~]# systemctl start mysqld

[root@mysql51 ~]# mysql -uroot -p123asd...A

**逻辑备份及恢复**

• 备份操作（完全）

– mysqldump -uroot -p密码 库名 > 路径 /xxx.sql

– mysqldump

例：

[root@client ~]# mysqldump -uroot -p654321 -A > /datadir/allbak.sql

//备份所有库

[root@client ~]# mysqldump -uroot -p654321 db5 > /datadir/db5.sql

//备份一个库

[root@client ~]# mysqldump -uroot -p654321 db5 t1 > /datadir/db5\_t1.sql

//备份库中的表

[root@client~]# mysqldump -uroot -p123456 -B db5 db55

> /datadir/twodb.sql //备份多个库

]# ls /datadir/\*.sql

[root@client ~]# cat /datadir/db5.sql

[root@client ~]# scp /datadir/db5\_t1.sql 192.168.4.51:/root/

[root@client ~]# scp /datadir/twodb.sql 192.168.4.51:/root/

• 恢复操作（完全）

– mysql -uroot -p密码 库名 < 路径/xxx.sql

例：

[root@mysql51 ~]# mysql -u root -p123asd...A < /root/twodb.sql

//恢复多个库，此时不需要指定库名，因在恢复文件中有建库的mysql命令

[root@mysql51 ~]# mysql -uroot -p123asd...A db5 < db5\_t1.sql

//恢复库中的表，此时需要指定库名，因在恢复文件中没有建库的命令，如果服务器上没有库，需要先建库，再恢复

mysql> create database db6; //连接数据库服务，创建库

mysql> use db6; //进入库

mysql> source /root/db5\_t1.sql; //恢复数据

mysql> show tables; //显示有t1表

• 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

**案例：每周一的半夜23：30 备份数据库服务器上的所有的数据到系统的/datadir目录下使用系统日期做为备份文件名称**

# vim /allbak.sh

.....

......

# chmod +x /allback.sh

# crontab -e -u root

### 实时增量备份 / 恢复

### 方法1： 使用binlog日志文件实现数库的增量备份和恢复

**binlog日志**

**简介：**

**二进制日志用途及配置方式**

|  |  |  |
| --- | --- | --- |
| **类型** | **用途** | **配置** |
| **二进制文件** | **记录所有更改数据的操作** | **log\_bin[=dir/name]**  **server\_id=数字**  **max\_binlog\_size=数字m** |

**分析binlog日志**

查看日志当前记录格式

mysql> show variables like 'binlog\_format';

**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

**启用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

[root@client ~]# vim /etc/my.cnf

server\_id=50 //指定id值

log\_bin //启用binlog日志

binlog\_format="mixed" //修改日志记录格式

[root@client ~]# systemctl restart mysqld

[root@client ~]# ls /var/lib/mysql

binlog相关文件

— 主机名-bin.index 记录已有日志文件名

— 主机名-bin.000001 二进制文件

[root@client ~]# mysqlbinlog /var/log/mysqld/ binlog查看日志文件命令

[root@client ~]# mysqlbinlog /var/lib/mysql/client-bin.000001

启用binlog日志文件时，指定存储位置和日志名

[root@client ~]# vim /etc/my.cnf

log\_bin=/logdir/asd

[root@client ~]# mkdir /logdir //创建日志目录

[root@client ~]# chown mysql /logdir //修改所有者为mysql

[root@client ~]# systemctl restart mysqld

[root@client ~]# ll -d /logdir/

屏幕输出：drwxr-xr-x. 2 mysql root 41 10月 20 11:49 /logdir/

[root@client ~]# ls /logdir/ //查看指定的目录下是否有日志文件

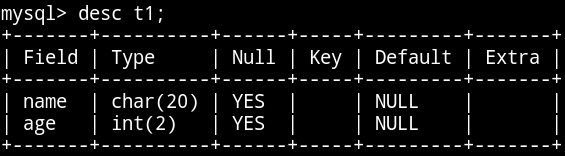
asd.000001 asd.index

验证binlog记录的信息

在数据库服务器上执行sql命令，查看日志文件是否记录了查询之外的sql命令

mysql> select \* from db5.t1;

mysql> desc t1;



插入操作：

mysql> insert into t1 values("bob",11);

mysql> insert into t1 values("jek",16);

mysql> insert into t1 values ("bin",444);

更改操作：

mysql> update db5.t1 set age=888 where age=444;

mysql> select \* from db5.t1; //查询操作

mysql> delete from db5.t1; //删除操作

查看日志文件中sql命令记录：

[root@client ~]# mysqlbinlog /logdir/asd.000001 | grep -i insert

有输出

[root@client ~]# mysqlbinlog /logdir/asd.000001 | grep -i update

有输出

[root@client ~]# mysqlbinlog /logdir/asd.000001 | grep -i delete

有输出

[root@client ~]# mysqlbinlog /logdir/asd.000001 | grep -i select

[root@client ~]# mysqlbinlog /logdir/asd.000001 | grep -i desc

[root@client ~]# mysqlbinlog /logdir/asd.000001 | grep -i show

**binlog恢复数据**

— 使用mysqlbinlog 提取历史SQL操作

—

• 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"

– --stop-datetime="yyyy-mm-dd hh:mm:ss"

– --start-position= 数字（起始位置偏移量）

– --stop-position= 数字（结束位置偏移量）

通过位置偏移量恢复数据

[root@client ~]# mysqlbinlog --start-datetime='2018-10-20 14:00:00' /logdir/asd.000001 //通过此命令查看binlog日志内容，

手动找出要恢复数据的起始位置偏移量与结束位置偏移量

[root@client ~]# mysqlbinlog --start-position=219

--stop-position=1268 /logdir/asd.000001 | mysql -uroot -p654321

[root@client ~]# mysql -uroot -p654321

mysql> select \* from db5.t1;

通过时间范围恢复数据

[root@client ~]# mysqlbinlog --start-datetime='2018-10-20 14:00:00' /logdir/asd.000001

[root@client ~]# mysqlbinlog --start-datetime="2018-10-20 14:05:35" --stop-datetime="2018-10-20 14:10:53" /logdir/asd.000001 | mysql -uroot -p654321

mysql> select \* from db5.t1;

在恢复过程中，可以不用指定起始位置，直接指定结束位置

手动创建新的binlog日志文件

默认日志容量大于500M 自动常见新的日志文件

显示当前数据库正在使用的日志文件及偏移量

mysql> show master status;



手动生成新的日志文件

mysql> flush logs;

[root@client ~]# mysql -uroot -p654321 -e 'flush logs';

[root@client ~]# mysqldump -uroot -p654321 --flush-logs db5 > /root/db5.sql

[root@client ~]# systemctl restart mysqld

**清理binlog日志**

1. 删除早于指定版本的 binlog日志

— purge master logs to “binlog文件名”;

mysql> purge master logs to 'asd.000001';

1. 删除所有binlog日志，重新建日志

— reset master;

mysql> reset master;

方法2： 使用innobackipex命令实现数据的增量备份和恢复

# MYSQL备份工具

**常用的MYSQL备份工具**

物理备份缺点

— 跨平台性差

— 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

**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

— 由专业组织Percona提供（改进MySQL分支）

主要含两个组件

— xtrabackup：C程序，支持InnoDB/XtraDB

— innobackupex：以Perl脚本封装xtrabackup,还支持MyISAM

**安装percona**

下载适配的RPM包并安装

[root@client ~]# rpm -ivh libev-4.15-1.el6.rf.x86\_64.rpm

# yum -y install percona-xtrabackup-24-2.4.7-1.el7.x86\_64.rpm

# rpm -ql percona-xtrabackup-24

**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 |
| --databases=”库名” //单个库  --databases=”库1 库2” //多个库  --databases=”库.表” //单个表 | |
| --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 | 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 |

**完全备份与恢复**

环境准备

1. 将默认存储引擎该为innodb

（innodb备份时可以增量备份，myisam即使执行增量的命令进行备份，可实际执行的还是完全备份）

1. 重新建库、表

mysql> create database db5;

mysql> create table db5.a(id int);

mysql> insert into db5.a values(110);

**备份**

[root@client ~]# innobackupex --user root --password 654321 /allbak

//完全备份，会在/allback目录下使用日期作为文件存储的子目录名

[root@client ~]# innobackupex --user root --password 654321 /allbak --no-timestamp //不用日期命令备份文件存储的子目录名

[root@client ~]# ls /allbak/

**恢复数据：1.准备恢复数据 --apply-log 2. 拷贝数据 --copy-back**

[root@client ~]# systemctl stop mysqld

[root@client ~]# rm -rf /var/lib/mysql //恢复时要求空的库目录

[root@client ~]# innobackupex --apply-log /allbak //准备恢复数据（回滚日志）

[root@client ~]# innobackupex --copy-back /allback //拷贝数据

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

//递归赋权限，给mysql赋予目录权限

[root@client ~]# systemctl start mysqld

[root@client ~]# mysql -uroot -p654321

mysql> select \* from db5.a

**增量备份和恢复——（增量备份时，必须先有一次备份，通常时完全备份）**

**增量备份：**备份上次备份后，所有新产生的数据

--incremental 目录 指定增量备份文件存储目录

--incremental-basedir=目录 指定上次备份文件存储目录

完全备份

[root@client ~]# innobackupex --user root --password 654321 /fullbak --no-timestamp

完全备份后，继续向表里写入新数据

[root@client ~]# mysql -uroot -p654321

mysql> insert into db5.a values(120);

增量备份

[root@client ~]# innobackupex --user root --password 654321 --incremental /new1dir --incremental-basedir=/fullbak --no-timestamp

第一次增量备份完成后，继续向表里写入新数据

mysql> insert into db5.a values(119);

第二次增量备份

[root@client ~]# innobackupex --user root --password 654321 --incremental /new2dir --incremental-basedir=/new1dir --no-timestamp

查看备份信息

[root@client ~]# ls /fullbak

[root@client ~]# ls /new1dir/

[root@client ~]# ls /new2dir/

xtrabackup\_checkpoints

[root@client ~]# cat /fullbak/xtrabackup\_checkpoints（数据的备份信息和备份类型）

from\_lsn = 0 日志序列号

**增量恢复**

1. 准备恢复数据—— --apply-log
2. 合并日志—— --redo-only
3. 拷贝数据—— --copy-back
4. 指定增量恢复的目录名——--incremental-dir=目录名

**删除数据**

[root@client ~]# systemctl stop mysqld

[root@client ~]# rm -rf /var/lib/mysql

[root@client ~]# mkdir /var/lib/mysql

**增量恢复数据**

[root@client ~]# cat /fullabk/xtrabackup\_checkpoints//查看恢复之前的信息

[root@client ~]# innobackupex --apply-log --redo-only /fullabk

[root@client ~]# cat /fullabk/xtrabackup\_checkpoints //查看恢复信息

[root@client ~]# innobackupex --apply-log --redo-only /fullabk/ --incremental-dir=/new1dir

[root@client ~]# cat /fullabk/xtrabackup\_checkpoints

[root@client ~]# innobackupex --apply-log --redo-only /fullabk/ --incremental-dir=/new2dir

[root@client ~]# cat /fullabk/xtrabackup\_checkpoints

**拷贝备份文件到数据库目录下**

[root@client ~]# innobackupex --copy-back /fullabk/

**修改所有者和组为mysql**

[root@client ~]# ls /var/lib/mysql

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

[root@client ~]# systemctl start mysqld

[root@client ~]# mysql -uroot -p654321

mysql> select \* from db5.a; //查看恢复的数据

mysql> insert into db5.a values(212); 继续存储新数据（写多条）

**备份新产生的数据**

[root@client ~]# innobackupex --user root --password 654321 --incremental /dir --incremental-basedir=/fullabk --no-timestamp

//恢复数据之后的第一次增量备份

[root@client ~]# cat /dir/xtrabackup\_checkpoints //查看备份信息

**恢复完全备份中的某个表**

在db5库下创建第2张表并写入记录

mysql> create table db5.b(id int);

mysql> insert into db5.b values(119); //插入多条

对db5库做完全备份

[root@client ~]# innobackupex --user root --password 123456 --databases='db5' /db5bak --no-timestamp

查看备份信息

[root@client ~]# ls /db5bak

删除db5库的a表

mysql> drop table db5.a;

单独恢复a表的步骤

--export 导出表信息

删除表空间 mysql> alter table 库名.表 discard tablespace;

导入表空间 mysql> alter table 库名.表 import tablespace;

1.手动创建表a且表结构要与删除时的表结构相同

mysql> create table db5.a(id int);

1. 删除a表的表空间文件

mysql> alter table db5.a discard tablespace;

mysql> select \* from db5.a;

[root@client ~]# ls /var/lib/mysql/db5

1. 导出表信息，并拷贝到数据库目录下

[root@client ~]# innobackupex --user root --password 123456 --databases="db5" --apply-log --export /db5bak

[root@client ~]# cp /db5bak/db5/a.{cfg,exp,ibd} /var/lib/mysql/db5/

//将恢复文件拷贝到数据库目录下

[root@client ~]# ls /var/lib/mysql/db5/a.\* //查看

[root@client ~]# chown mysql:mysql /var/lib/mysql/db5/a.\* //更该归属

1. 导入表信息 并删除表信息文件

mysql> alter table db5.a import tablespace;

mysql> system rm -rf /var/lib/mysql/db5/a.cfg //恢复之后删除表恢复文件建

mysql> system rm -rf /var/lib/mysql/db5/a.exp

mysql> system ls /var/lib/mysql/db5/a.\* //查看

/var/lib/mysql/db5/a.frm（表结构） /var/lib/mysql/db5/a.ibd（表空间文件）

1. 查看表记录

mysql> select \* from db5.a; //查看表内容是否恢复