**mysqldump备份/恢复**

数据库备份描述

数据备份方式

物理备份

冷备：cp tar

逻辑备份

mysqldump

mysql

数据库备份策略

完全备份：备份所有数据

增量备份：备份上次备份后，所有新产生的数据

差异备份：备份完全备份后，所有新产生的数据

生产环境：

备份频率：取决于数据量产生的频率

备份策略：完全+增量 完全+差异

备份时间：数据库服务器不忙，因为有些备份操作要加写锁

备份文件名：日期\_xx.sql

存储空间：lvm + raid

物理备份

物理备份及恢复（innodb不适合物理备份，适合mysiam）

备份操作

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/\*



恢复操作

cp -rp 备份目录/文件名 /var/lib/mysql

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







逻辑备份（执行备份操作时，根据备份的库表产生对应的sql命令，把sql命令存储到指定的文件里）

逻辑备份与恢复

备份操作：mysqldump -uroot -p密码 库名 > 路径/xxx.sql



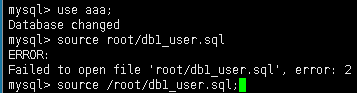






恢复操作：mysql -uroot -p密码 库名 < 路径/xxx.sql





库名表示方式：--all-databases或-A 所有库

数据库名 单个库

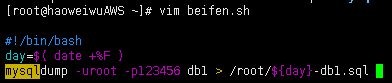
数据库名 表名 单张表

-B 数据库1 数据库2 多个库

注意事项

无论备份还是恢复，都要验证用户权限

备份脚本



**实时增量备份/恢复**

binlog简介

binlog日志（二进制日志，记录除查询之外的sql命令）概述

二进制日志用途及配置方式

|  |  |  |
| --- | --- | --- |
| 类型 | 用途 | 配置 |
| 二进制日志 | 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（默认500M） |

使用binlog日志

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

vim /etc/my.cnf

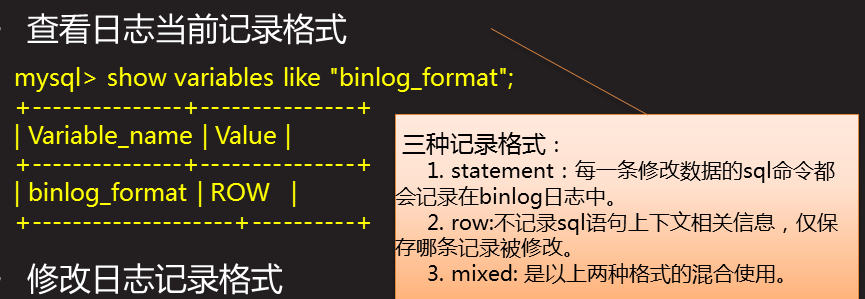
[mysqld]

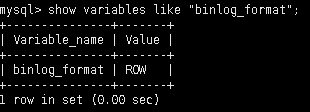
log\_bin //启用binlog日志

server\_id=100 //指定id值

systemctl resstart mysqld

查看日志当前记录格式





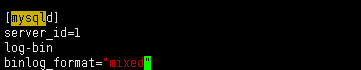
修改日志记录格式

vim /etc/my.cnf

[mysqld]

binlog\_format=mixxed

systemctl restart mysqld



binlog相关文件

主机名-bin.index 记录已有日志文件名

主机名-bin.000001 第一个二进制日志（大于500M会生成第二个）

主机名-bin.000002 第二个二进制日志

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’

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;

分析binlog日志

使用mysqlbinlog工具

格式：mysqlbinlog [选项] binlog日志文件名

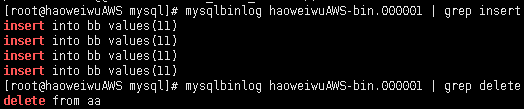
常用选项

--start-datetime=”yyyy-mm-dd hh:mm:ss”

--stop-datetime=”yyyy-mm-dd hh:mm:ss”

--start-position=数字

--stop-position=数字



binlog恢复数据

利用binlog恢复数据

基本思路

使用Mysqlbinlog提取历史sql操作

通过管道交给mysql命令执行

应用示例

执行第一份binlog所记录的更改操作

cd /var/lib/mysql

mysqlbinlog mysql-bin.000001 | mysql -uroot -p123456