# 前言

我们试着想一想，在生产环境中什么最重要？如果我们服务器的硬件坏了可以维修或者换新，软件问题可以修复或重新安装，但是如果数据没了呢？这可能是最恐怖的事情了吧，我感觉在生产环境中应该没有什么比数据更为重要。那么我们该如何保证数据不丢失、或者丢失后可以快速恢复呢？

# 为什么需要备份数据？

其实在前言中也大概说明了为什么要备份数据，但是我们还是应该具体了解一下为什么要备份数据。

在生产环境中我们数据库可能会遭遇各种各样的不测从而导致数据丢失，大概分为以下几种：

1. 硬件故障
2. 软件故障
3. 自然灾害
4. 黑客攻击
5. 误操作（占比最大）

所以，为了在数据丢失之后能够恢复数据，我们就需要定期的备份数据，备份数据的策略要根据不同的应用场景定制，大致有几个参考数值，我们可以根据这些数值从而定制符合特定环境中的数据备份策略：

1. 能够容忍丢失多少数据
2. 恢复数据需要多长时间
3. 需要恢复哪些数据

# 数据的备份类型（备份目标）

数据的备份类型格局其自身的特性主要分为以下几组

## 完全备份

完全备份指的是备份整个数据集（即整个数据库）

## 部分备份

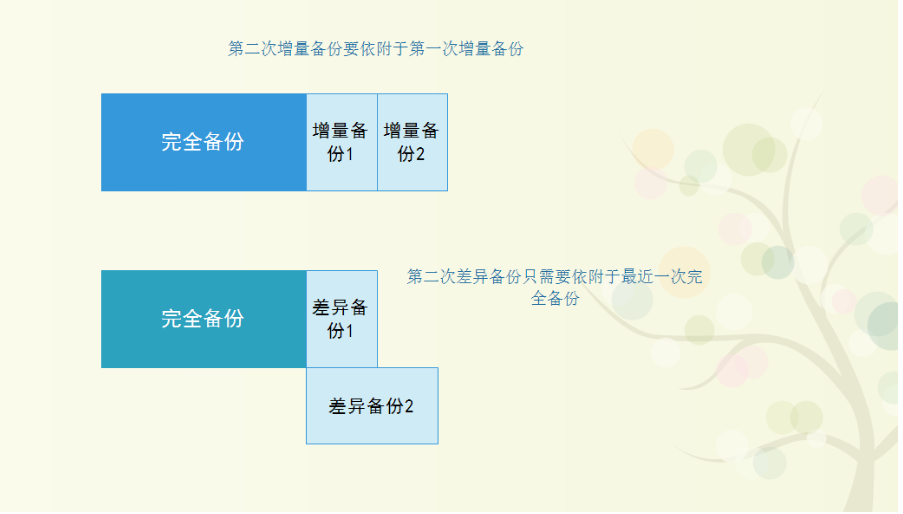
部分备份指的是备份部分数据集（例如：只备份一张表）。而部分备份又分为以下两种

### 增量备份

增量备份指的是备份自上一次备份以来变化的数据（特点：节约空间、还原麻烦）

### 差异备份

差异备份指的是备份自上一次完全备份以来变化的数据（特点：浪费空间、还原简单）



# MySQL备份数据的方式

在MyQSL中我们备份数据一般有三种方式

## 热备份

数据库进行备份时，数据库的读写操作均不受影响

## 温备份

数据库进行备份时，数据库的读操作可以执行，但是写操作不能执行

## 冷备份

数据库进行备份时，数据库要下线，即不能执行读写操作

## MySQL中进行不同方式的备份还要考虑存储引擎是否支持

### MyISAM

热备×

温备√

冷备√

### InnoDB

热备√

温备√

冷备√

## 备份手段

我们在考虑完数据在备份时数据库的运行状态之后，还需要考虑对于MySQL数据库中数据的备份手段

### 物理备份

一般就是通过tar,cp等命令直接打包赋值数据库的数据文件达到备份的效果

### 逻辑备份

一般就是通过特定工具从数据库中导出数据并另存备份（会丢失数据精度）

# 备份需要考虑的问题

定制备份策略前，我们还需要考虑一些问题

## 我们要备份什么？

1. 数据
2. 二进制文件，InnoDB事务日志
3. 代码（存储过程、存储函数、触发器、事件调度器）
4. 服务器配置文件

## 备份工具

### mysqldump

逻辑备份工具，适用于所有的存储引擎，支持温备、完全备份、部分备份、对于InnoDB存储引擎支持热备

### cp tar等归档复制工具

物理备份工具，适用于所有的存储引擎，冷备、完全备份、部分备份

### lvm2-snapshot

几乎热备，借助文件系统管理工具进行备份

### mysqlhotcopy

名不副实的一个工具，几乎热备，仅支持MyISAM存储引擎

### xtrabackup

一款非常强大的InnoDB/XtraDB热备工具，支持完全备份、增量备份，由percona提供

# 设计合适的备份策略

针对不同的场景下，我们应该制定不同的备份策略对数据库进行备份

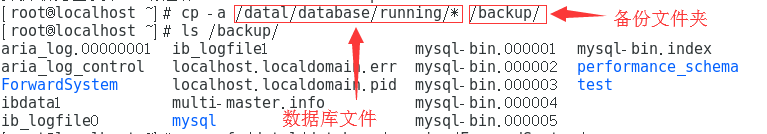
## 直接cp tar 赋值数据库文件

数据量小；适用于所有的存储引擎；冷备、完全备份、部分备份。

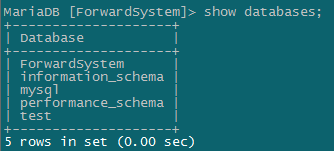
### 向数据库施加读锁

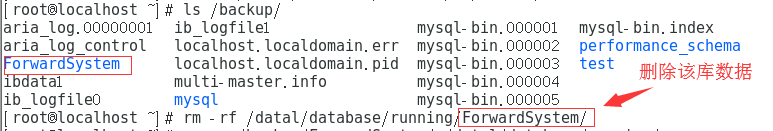


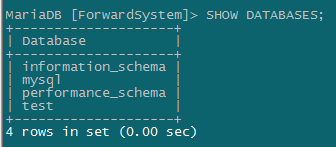
### 备份数据文件



### 模拟数据丢失

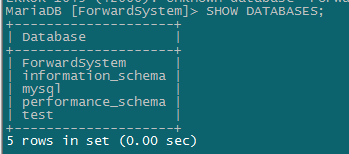






### 数据恢复





## mysqldump+复制BIN LOGS

数据量还行；先使用mysqldump对数据库进行完全备份，然后定期备份BINARY LOG达到增量备份的效果；二进制日志需要在mysql配置文件[mysqld]中添加log\_bin=mysql-bin开启。

### mysqldump

#### 命令介绍

mysqldump是一个客户端的逻辑备份工具，可以生成一个重现创建原始数据库和表的SQL语句，可以支持所有的存储引擎，对于InnoDB支持热备。

#### 基本语法格式

shell> mysqldump [options] dbName [tbName1 …] #恢复需要手动CREATE DATABASE;

shell> mysqldump [options] –databases dbName #恢复不需要手动CREATE DATABASE;

shell> mysqldump [options] –all-databases #恢复不需要手动CREATE DATABASE;

/\* 1.导出整个数据库 \*/

mysqldump -u用户名 -p 数据库名 > target.sql;

/\* 2.导出一个表（包括表结构和数据） \*/

mysqldump -u用户名 -p 数据库名 表名 > target.sql;

/\* 3.只导出表结构 \*/

mysqldump -u用户名 -p -d 数据库名 表名 > target.sql;

#### 其它选项

-E, --events：备份事件调度器

-R, --routines：备份存储过程和存储函数

--triggers：备份表的触发器；--skip-triggers

--master-date[=value]

1：记录为CHANGE MASTER TO 语句、语句不被注释

2：记录为注释的CHANGE MASTER TO 语句

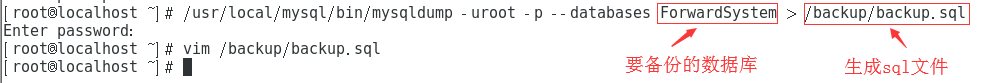
基于二进制还原只能全库还原

--flush-logs：日志滚动

锁定表完成后执行日志滚动

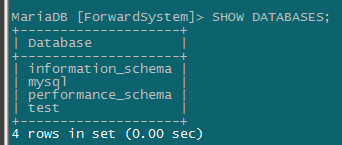
#### 使用mysqldump备份数据库

1. 备份数据

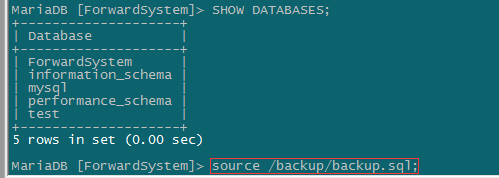


1. 模拟数据丢失





1. 利用mysqldump备份生成的sql文件，恢复完全备份数据。



### BIN-LOG日志

#### 基本定义

二进制日志，记录对数据发生或潜在发生更改的SQL语句，并以二进制形式保存在磁盘中。

#### 作用

1. 可以用来查看数据库的变更历史（具体的时间点所有的SQL操作）
2. 数据库增量备份和恢复（增量备份和基于时间点的恢复）
3. MySQL的复制（主主数据库的复制、主从数据库的复制）

#### 文件位置

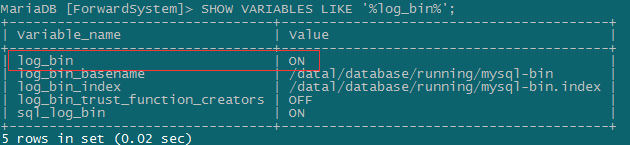
默认存放位置为数据库文件所在目录下

#### 文件的命名方式

文件名为hostname-bin.xxxxx（重启mysql一次将会自动生成一个新的bin-log，使用命令：FLUSH LOGS;也可以生成新的bin-log）

#### 状态的查看

mysql> SHOW VARIABLES LIKE ‘%log\_bin%’;



#### 开启bin-log日志

在MySQL配置文件中添加：log-bin=mysql-bin

然后重启下MySQL

#### 打开bin-log日志文件

用VIM打开二进制日志是乱码的，所以我们需要使用下面的命令进行打开：

例：/usr/local/mysql/bin/mysqlbinlog --no-defaults /backup/mysql-bin.000005

#### 常用命令

FLUSH LOGS; 会多一个最新的bin-log日志。

SHOW MASTER STATUS; 查看最后一个bin-log日志的相关信息。

RESET MASTER; 清空所有的bin-log日志。

#### 从BIN-LOG日志恢复数据

1. 恢复命令的语法格式：

mysqlbinlog mysql-bin.000xxx [options] | mysql –u用户名 –p 数据库名



1. 常用[options]解释：

--start-position=875 起始pos点为875

--stop-position=999 结束pos点为999

--start-datetime=”2019-1-4 16:34:45” 起始时间点

--stop-datetime=” 2020-1-4 16:34:45” 结束时间点

--database=dbName 指定只恢复dbName数据库

1. 不常用选项

-u --user=name 连接到远程主机的用户名

-p --password[=name] 连接到远程主机的密码

-h –host=name 从远程主机上获取binlog日志

--read-from-remote-server 从某个MySQL服务器上读取binlog日志

**\*实际是将读出的binlog日志内容，通过管道符传递给mysql命令。**

**\*所谓的恢复，就是让mysql将保存在binlog日志中指定段落区间的sql语句逐个重新执行一次罢了**

## lvm2快照+复制BIN LOGS

数据量一般，而又不过分影响业务运行；使用lvm2的快照对数据文件进行备份，而后定期备份BINARY LOG达到增量备份的效果

## xtrabackup

数据量很大，而又不过分影响业务运行；使用xtrabackup进行完全备份后，定期使用xtrabackup进行增量备份或差异备份