# Mysql

## MYSQL用户授权

grant 权限 on 数据库.\* to 用户名@登录主机 identified by “密码”

grant replication slave on \*.\* to ‘mysql12’@’192.168.95.12’ identified by ‘mysql12’;

# 数据备份

## 数据备份方式：

### 完全备份：

备份整个数据集,即备份整个数据库;

### 部分备份：

备份部分数据集，如只备份一个表;

# Mysql数据备份：

## 相关资料：

Mysql备份的解决方案：

<https://blog.csdn.net/zizhilianqiu/article/details/89297596>

Mysql备案方式：

<https://www.jianshu.com/p/25cca24a54cf>

备份操作：

<https://blog.csdn.net/wu631464569/article/details/78242038>

全备份：

<https://www.jianshu.com/p/e80041ea9d93>

## Mysql的备份方式：

### 热备份

当数据库进行备份时, 数据库的读写操作均不是受影响；

### 温备份

当数据库进行备份时, 数据库的读操作可以执行, 但是不能执行写操作

### 冷备份

当数据库进行备份时, 数据库不能进行读写操作, 即数据库要下线

## Mysql备份需要考虑的问题

### Mysql存储引擎是否支持

|  |  |  |  |
| --- | --- | --- | --- |
| 存储引擎 | 热备份 | 温备份 | 冷备份 |
| MyISAM | 不支持 | 支持 | 支持 |
| InnoDB | 支持 | 支持 | 支持 |

### Mysql数据库中数据的备份方式

#### 物理备份

通过tar,cp等命令直接打包复制数据库的数据库文件，达到备份的效果;

#### 逻辑备份

通过特定的工具从数据库中导出数据并另存备份，逻辑备份会丢失数据精度；

### 我们需要备份什么数据

### 备份工具

#### mysqldump

**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

**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

**lvm2 snapshot**: 几乎热备, 借助文件系统管理工具进行备份

#### mysqlhotcopy

mysqlhotcopy:名不副实的的一个工具, 几乎冷备, 仅支持**MyISAM存储引擎**

#### xtrabackup

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

### 选择合适的备份策略

#### 直接cp,tar复制数据库文件

1.数据量较小, 直接复制数据库文件;

#### mysqldump+复制BIN LOGS

如果数据量还行, 可以先使用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

#### lvm2快照+复制BIN LOGS

如果数据量一般, 而又不过分影响业务运行, 可以使使用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

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

## 使用cp进行备份

### 查看数据库的信息

#显示数据库

SHOW DATABASES;

#查询当前看库中的表

SHOW TABLES;

### ****向数据库施加读锁****

**当mysql为一个表开启读锁的时候,其他进程包括进程本身没有权利去修改这张表的内容,也没有权利更新,但是可以读取表里面的内容;**

#向所有表施加读锁

FLUSH TABLES WITH READ LOCK;

### ****备份数据文件****

#创建文件夹存放备份数据库文件

mkdir /backup

#拷贝源数据文件

cp -a /var/lib/mysql/\* /backup

#查看目录下的文件

ls /backup

### 4.执行解锁

**unlock tables**

### 5.数据恢复

#删除文件

rm -rf /var/lib/mysql/\*

#将备份的数据文件拷贝回去

1. cp -a /backup/\* /var/lib/mysql/
2. 重启MySQL#重新连接数据并查看mysql

## 使用mysqldump+复制BINARY LOG备份

通过mysqldump进行一次完全备份, 再修改表中的数据, 然后再通过binary log进行恢复 二进制日志需要在mysql配置文件中添加 log\_bin=on 开启；

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

官方文档：<https://dev.mysql.com/doc/refman/5.7/en/mysqldump.html>

### #基本语法格式

**mysqldump [options] db\_name [tbl\_name ...]**  恢复需要手动

m**ysqldump [options] --databases db\_name ...**  恢复不需要手动创建数据库

**mysqldump [options] --all-databases**  恢复不需要手动创建数据库

其他选项:

-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

### ****查看数据库的信息****

### ****使用****mysqldump****备份数据库****

#查看当前二进制文件的状态, 并记录下position的数字

**mysql -uroot -p -e 'SHOW MASTER STATUS'**

#备份数据库到backup.sql文件中

**mysqldump --all-databases --lock-all-tables > backup.sql**

#创建一个数据库

CREATE DATABASE TEST1;

#记下现在的position

SHOW MASTER STATUS;

cp /var/lib/mysql/mysql-bin.000003 /root #备份二进制文件

#停止MySQL

## 使用MYSQL主从复制实现数据热备份

### 主从复制的优势

1.使用主从复制，让主库负责写，从库负责读，可提高程序的稳定性;

2.可以做数据热备份

3.架构拓展。

相关资料:

<https://blog.51cto.com/u_13589255/2095858>

<https://www.jianshu.com/p/b0cf461451fb>

### 什么是MYSQL的主从复制

MySQL 主从复制是指数据可以从一个MySQL数据库服务器主节点复制到一个或多个从节点。

MySQL 默认采用异步复制方式，这样从节点不用一直访问主服务器来更新自己的数据，数据的更新可以在远程连接上进行，从节点可以复制主数据库中的所有数据库或者特定的数据库，或者特定的表。

### MYSQL的复制原理：

master服务器将数据的改变记录二进制binlog日志，当master上的数据发生改变时，则将其改变写入二进制日志中；

slave服务器会在一定时间间隔内对master二进制日志进行探测其是否发生改变，如果发生改变，则开始一个I/OThread请求master二进制事件

### MYSQL主从形式

#### 1.一主一从

#### 2.主主复制

#### 3.一主多从

#### 4.多主一从

#### 5.级联复制

相关资料

<https://zhuanlan.zhihu.com/p/96212530>

<https://blog.csdn.net/zai_xia/article/details/90379016>

<https://www.cnblogs.com/xihuineng/p/10438969.html>

## MYSQL的主从复制

### 准备工作

1. 准备两台机器，安装MYSQL

### 基本步骤

#### 配置主数据库(master)

##### 1.1修改主数据库配置文件

#设置主服务 的ID (id可以自己随便设置但是要保证和slave的id不一样)

server-id=200​

innodb\_flush\_log\_at\_trx\_commit=2

#(参数的含义如下)​

# 0：如果innodb\_flush\_log\_at\_trx\_commit的值为0,log buffer每秒就会被刷写日志文件到磁盘，提交事务的时候不做任何操作（执行是由mysql的master thread线程来执行的。

# 主线程中每秒会将重做日志缓冲写入磁盘的重做日志文件(REDO LOG)中。不论事务是否已经提交）默认的日志文件是ib\_logfile0,ib\_logfile1

# 1：当设为默认值1的时候，每次提交事务的时候，都会将log buffer刷写到日志。

# 2：如果设为2,每次提交事务都会写日志，但并不会执行刷的操作。每秒定时会刷到日志文件。要注意的是，并不能保证100%每秒一定都会刷到磁盘，这要取决于进程的调度。

# 每次事务提交的时候将数据写入事务日志，而这里的写入仅是调用了文件系统的写入操作，而文件系统是有 缓存的，所以这个写入并不能保证数据已经写入到物理磁盘

# 默认值1是为了保证完整的ACID。当然，你可以将这个配置项设为1以外的值来换取更高的性能，但是在系统崩溃的时候，你将会丢失1秒的数据。

# 设为0的话，mysqld进程崩溃的时候，就会丢失最后1秒的事务。设为2,只有在操作系统崩溃或者断电的时候才会丢失最后1秒的数据。InnoDB在做恢复的时候会忽略这个值。

# 总结

# 设为1当然是最安全的，但性能页是最差的（相对其他两个参数而言，但不是不能接受）。如果对数据一致性和完整性要求不高，完全可以设为2，如果只最求性能，例如高并发写的日志服务器，设为0来获得更高性能​

#开启binlog 志同步功能

sync\_binlog=1 ​

#binlog 日志文件名

log-bin=mysql-bin-200​

# 这个表示只同步某个库 (如果没有此项，表示同步所有的库)

binlog-do-db=xxxx

##### 1.2重启主数据库

##### 1.3登录主数据库,创建用于从服务器同步数据使用的帐号

GRANT REPLICATION SLAVE ON \*.\* TO 'slave'@'%' identified by 'slave';

FLUSH PRIVILEGES;

##### 1.4获取主服务器的二进制日志信息

SHOW MASTER STATUS;

#### 配置从数据库(slave)

##### 2.1修改从数据库配置文件

server-id=201

innodb\_flush\_log\_at\_trx\_commit=2

sync\_binlog=1

log-bin=mysql-bin-201

##### 2.2重启从数据库

##### 2.3进入从数据库,设置连接到master主服务器;

change master to master\_host='192.168.1.200',master\_user='mark' ,master\_password='12345', master\_log\_file='mysql- bin-200.000004' ,master\_log\_pos=671;

参数说明:

master\_host: 主机的ip

master\_user : 主机授权的用户.

master\_password : 主机授权时候填写的密码

master\_log\_file : 主机show master status;中的File

master\_log\_pos: 主机show master status;中的Position.

##### 2.4开启同步，查看同步状态

start slave;

show slave status \G;

#### 3.测试主从同步

相关资料：

<https://www.cnblogs.com/phpstudy2015-6/p/6485819.html#_label3>

<https://blog.csdn.net/weiwei101117/article/details/52038330>

<https://blog.csdn.net/wang55812526/article/details/38943621>

https://blog.csdn.net/qq\_33196814/article/details/81513907

## MYSQL的主主复制

相关资料：<https://blog.51cto.com/yangrong/1569700>

# Oracle数据备份

## 相关资料

https://www.cnblogs.com/klb561/p/10828935.html