**用户授权**

-- 默认数据库管理员有授权权限

-- 安装mariadb就有了连接数据库命令

1. 什么是用户授权: 添加新的连接用户,设置密码及访问权限
2. 授权命令格式: grant

-- grant 权限列表 on 库名.表名 to 用户@”客户端地址”

identified by “密码”

with grant option; ----授权权限,可选项

权限列表:

- all: #所有权限

- usage: #无权限,只能连接

- select ,update ,insert #个别权限

- select ,update(字段1,...,字段n) #指定字段权限

库名:

- \*.\* #所有权限

- 库名.\* #一个库下的所有表

- 库名.表名 #一张表

客户端地址:

- localhost: #数据库服务器本机

- %: #所有主机,只要能跟数据库服务器ping通就能连接

- 192.168.4.%: #网段内所有主机

3.相关命令

***权限的 增|删|改|查***

**· 授权用户及权限**

- grant 权限列表 on 库名.表名 to 用户@”客户端地址”

Identified by “密码”

with grant option; #授权权限,可选项

**· 查看权限**:

- show grants; #显示当前用户权限

- show grants for 用户名@”地址”; #管理员查看用户权限(普通用户此命令无权)

**· 修改权限**

- 通过修改表的记录修改权限 **改完需刷新表**

**· 撤销权限**:revoke

- revoke 权限列表 on 库名.表 from 用户名@”地址”;

**· 删除授权用户**:

- drop user 用户名@”地址”; #必须有root权限

***修改密码***

**·** 修改密码:

#授权用户,连接后自改密:

- set password=password(“密码”); #**password()为加密算法,必加**

#管理员改授权用户密码:

- set password for 用户名@”地址”=password(“密码”);

#命令行修改密码:

- ]#mysqladmin -uroot -p password “新密码”

1. 授权库mysql(记录授权信息管理)

mysql库:

与授权相关的表:

- user: 记录已有的授权**用户**及权限

- db: 记录已有授权用户对**数据库**的访问权限

- tables\_priv: 记录已有授权用户对**表**的访问权限

- columns\_priv: 记录已有授权用户对**字段**的访问权限

**#可以操作表记录,查看|更新 授权用户的权限**

**#用命令获取权限信息是通过表来获取的**

**#没有名字,所有权限或没设置权限**

虚拟库:

- **information\_schema**: 一个假库,不存在的库,系统内存的映射,只能看,没有任何

操作权限

1. 撤销权限:revoke

- revoke 权限列表 on 库名.表 from 用户名@”地址”;

1. 删除授权用户:drop

- drop user 用户名@”地址”;

1. 管理root密码
2. 恢复密码

思路: 启动mysql时绕过密码项, 进入mysql后修改root密码,重启使用新密码登

录

]#systemctl stop mysqld #关闭mysql

]#vim /etc/my.cnf #修改配置,绕过密码项

[mysqld]

**skip-grant-tables**

#validate\_password\_policy=0

#validate\_password\_length=6

]#systemctl start mysqld

]#mysql #通过修改表记录修改root密码

>update mysql.user set **authentication\_string**=password(“新密码”)

->where user=”root” and host=”localhost”;

>exit

]#vim /etc/my.cnf #恢复配置

[mysqld]

#skip-grant-tables

validate\_password\_policy=0

validate\_password\_length=6

]#systemctl restart mysqld

2.修改密码(重置密码)

命令行修改密码:

]#mysqladmin -uroot -p password “新密码”

##########################################################################

**二. 数据完全备份**

1.数据备份的目的

2.数据备份方式

* 物理备份: cp , tar #直接拷贝库和表对应的文件

备份: ]#cp **-r** /var/lib/mysql /root/**mysql.bak** **#要以bak结尾**

]#scp **-r** /root/mysql.bak [root@192.168.4.51:/root/](mailto:root@192.168.4.51:/root/)

恢复: ]#rm -rf /var/lib/mysql

]#cp -r mysql.bak /var/lib/mysql

]#chown -R mysql:mysql /var/lib/mysql **#修改权限**

]#systemctl restart mysqld

* 逻辑备份:

**使用软件自带的备份程序(备份命令)对数据备份,在执行备份命令时,会根据已有的数据生成对应的命令,把命令存放到指定的备份文件里,恢复时,执行备份文件里的命令恢复数据(备份文件里存的是命令)**

-- mysqldump #备份命令

-- myslq #恢复命令

#############################################################################

**逻辑备份**

**1.数据备份策略**

-- 完全备份: 备份所有数据( 1张表, 1个库, 1台服务器)

-- 差异备份: 备份完全次备后,所有新产生的数据(可能会重复)

-- 增量备份: 备份上次备份后,所有新产生的数据(都是新的)

**一. 完全备份&完全恢复**

备份后的文件为创建数据的命令

备份命令格式:

-- mysqldump -uroot -p密码 库名 > 目录/xxx.**sql**

恢复命令格式:

-- mysql -uroot -p密码 [库名] < 目录/xxx.sql

#恢复单个库或表时,需要指库名(单库,表时备份文件没里有创建库和切换库

的命令)不指定报错:No database selected

#恢复在备份文件里的多个库或表时,不需要指定库

**·** 备份时库名表示方式

-- -A 或 --all-databases

-- 数据库名

-- 数据库名 表名

-- -B 数据库1 数据库2

?????????

#################################################################################

**二.数据增量备份**

***binlog日志***

***1.什么是binlog日志***

-- 也称为二进制日志

-- 记录查询之外的所有SQL命令(只要发生变化就记录)(sql历史命令记录本)

-- 主从同步必备条件

-- Mysql**服务日志**的一种

***2.启动日志***

修改配置文件

]#vim /etc/my.cnf

[mysqld]

server\_id=数字 #id:编号,(0-255)

log-bin[=目录名/文件名] #(底 ”\_” 或 ”-” 都行),日志存放路径及名字

max\_binlog\_size=数值m #日志文件容量,默认1G(可不加此项)

]#systemctl restart mysqld

查看日志文件名

> show master status; #查看binlog日志状态

--------------------------------+-------------+---

| File | Position |

--------------------------------+-------------+----

| localhost-bin.000001 | 154 |

------------------------------- +-------------+---

#默认格式: 主机名-bin. 6位数编号

#自定义名称: 文件名.6位数编号

#154: 默认起始为154

日志文件存放路径

#若log-bin后没加路径,默认存放在数据库目录/var/lib/mysql/

localhost-bin.000001 --->日志文件1

localhost-bin.index ---> 索引文件(里面为所有日志文件名)

1. ***生成新的日志***

4种方法:

1. 重启mysql
2. mysql> flush logs
3. ]#mysql -uroot -p密码 -e ”flush logs”
4. 备份后刷新日志

]#mysqldump -uroot -p密码 **--flush logs** 库名 > 目录/xxx.sql

1. ***清除日志***

删除指定编号之前的日志:

-- purge master logs to "日志名";

删除所有日志,重新建日志:

-- reset master;

154

#################################################################################

**恢复数据**

***1.查看日志***

]# mysqlbinlog [选项] 目录/日志名 #***一堆乱码***

选项:

根据时间点:

--start-datetime=”yyyy-mm-dd hh:mm:ss” #起始时间

--stop-datetime=”yyyy-mm-dd hh:mm:ss” #结束时间

根据偏移量:

--start-position=数字 #起始偏移量

--stop-position=数字 #结束偏移量

**增量备份**

1. 修改日志格式

三种记录方式: statement(报表模式)

row(行模式)(**默认**)

mixed(混合模式)

查询记录方式:

> show variables like "**binlog\_format**";

修改记录方式:

]#vim /etc/my.cnf

[mysqld]

binlog\_format="mixed"

重启mysql

1. 日志格式翻译

# at 293

# at 325

#190809 10:10:34 server id 50 end\_log\_pos 325 CRC32 0xe66b6527 Intvar

SET INSERT\_ID=26/\*!\*/;

#190809 10:10:34 server id 50 end\_log\_pos 446 CRC32 0xf026b056 Query thread\_id=3 exec\_time=0 error\_code=0

SET TIMESTAMP=1565316634/\*!\*/;

insert into db3.t1(name,password) values("yaya","x")

/\*!\*/;

# at 446

#190809 10:10:34 server id 50 end\_log\_pos 477 CRC32 0x50c8c94c Xid = 9

COMMIT/\*!\*/; #命令输入完时敲的回车

1. 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

• 命令格式

–mysqlbinlog [选项] 日志文件 | mysql -uroot -p密码

可加选项来控制恢复的前后范围