

第1章 启动与关闭MySQL

1.启动方式

常规启动:

```
1 service mysqld start
2 systemctl start mysqld
3 /etc/init.d/mysqld start
```

前台启动:

如果作为调试可以直接使用命令启动，这是会作为前台启动将日志都打印到屏幕上

```
1 mysqld
2 mysqld_safe
```

2.mysqld_safe和mysqld区别

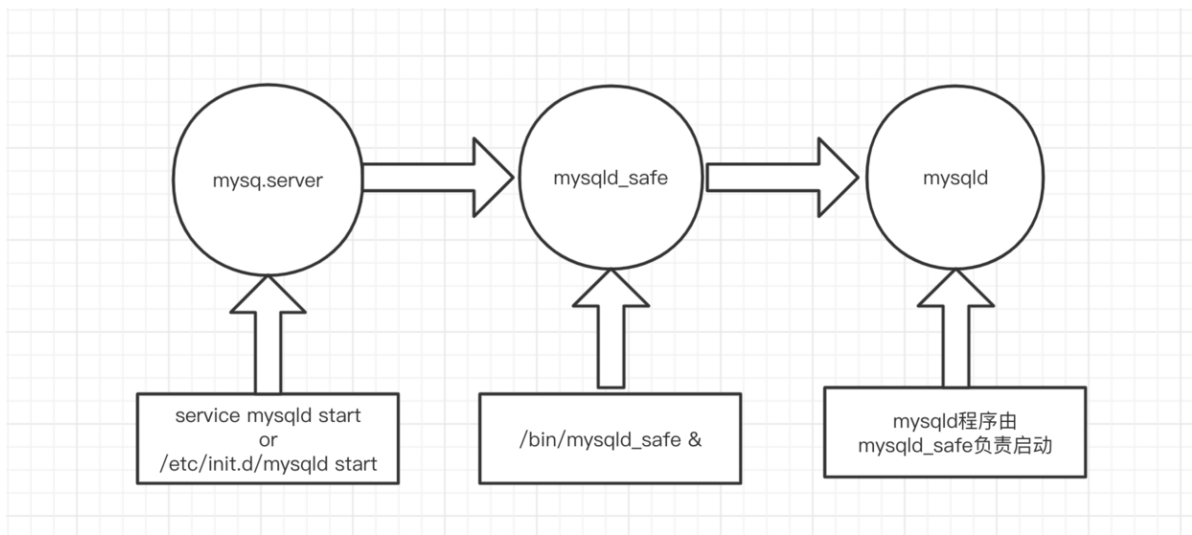
mysqld_safe作用:

1. 官方提供的启动脚本最终会调用mysqld_safe命令脚本，然后mysqld_safe脚本又会调用mysqld主程序启动MySQL服务。
2. 最终都是由mysqld启动，mysqld_safe可以看做是mysqld的守护进程。
3. mysqld_safe会启动并监视mysqld，如果mysqld发生意外错误可以重启服务。
4. mysqld_safe启动可以将mysqld的错误消息发送到数据目录中的host_name.err文件
5. 可以读取的配置部分[mysqld],[server],[mysqld_safe]，为了兼容mysqld_safe也会读取[safe_mysqld]中的配置
6. 调用的mysqld是可以在[mysqld_safe]中用-mysqld, --mysqld-version指定

mysqld作用:

1. mysqld是mysql的核心程序，用于管理mysql的数据库文件以及用户的请求操作。
2. mysqld可以读取配置文件中的[mysqld]的部分

启动流程示意图:



3.调试方式

调试命令:

```
1 | mysql
2 | mysqld_safe
```

应用场景:

1. 调试数据库启动故障。
2. 临时的启动参数应用。

4.关闭方式

常规关闭:

```
1 | systemctl stop mysqld
2 | service mysqld stop
3 | /etc/init.d/mysqld stop
```

命令关闭:

```
1 | mysql -uroot -p123456
2 | mysql> shutdown;
3 |
4 | mysql -uroot -p123456 -e 'shutdown;'
```

不建议关闭方法:

```
1 | kill PID
2 | pkill mysqld
3 | killall mysqld
```

强烈不建议:

```
1 | kill -9 PID
```

第2章 配置管理

1.配置方法

- 1 | 配置文件 : 主要方法
- 2 | 命令行参数 : 特殊场景
- 3 | 源码编译安装: 应用不多

2.配置优先级

- 1 | 命令行 > 配置文件 > 编译
- 2 | 例如: port
- 3 | 配置文件: port=3306 2
- 4 | 命令行: port=3307 1
- 5 | 源码编译: port=3308 3

3.配置文件位置

- 1 | [root@db-51 ~]# mysqld --help --verbose |grep my.cnf
- 2 | /etc/my.cnf
- 3 | /etc/mysql/my.cnf
- 4 | /usr/local/mysql/etc/my.cnf
- 5 | ~/.my.cnf

实验: 4个配置文件都添加port, 而且参数不一样, 以哪个配置为准?

- 1 | 会以最后一个配置为准

4.手动指定配置文件

手工设定默认配置文件

- 1 | --defaults-file=/opt/my.cnf

启动命令

- 1 | [root@db01 ~]# mysqld_safe --defaults-file=/opt/my.cnf &
- 2 | [root@db01 ~]# mysqld --defaults-file=/opt/my.cnf &

5.编写自定义启动脚本

- 1 | cat > /etc/systemd/system/mysqldd.service <<EOF
- 2 | [Unit]
- 3 | Description=MySQL Server
- 4 | Documentation=man:mysqld(8)
- 5 | Documentation=http://dev.mysql.com/doc/refman/en/using-systemd.html
- 6 | After=network.target
- 7 | After=syslog.target
- 8 | [Install]
- 9 | WantedBy=multi-user.target

```

10 [Service]
11 User=mysql
12 Group=mysql
13 ExecStart=/opt/mysql/bin/mysqld --defaults-file=/opt/my.cnf
14 LimitNOFILE = 5000
15 EOF

```

6.配置文件格式

```

1 [mysqld] # 服务端标签
2 user=mysql # MySQL内置管理用户
3 basedir=/data/app/mysql # 软件目录
4 datadir=/data/3306/data # 数据目录
5 socket=/tmp/mysql.sock # socket文件生成目录
6 port=3306 # 端口号
7 server_id # 主机编号（主从使用）
8
9 [mysql]
10 socket=/tmp/mysql.sock # mysql 连接数据库自动读取的socket位置
11
12 [标签项]
13 服务端标签：
14 [server] 所有服务端的程序
15 [mysqld] mysqld 程序运行时读取的配置
16 [mysqld_safe] mysqld_safe 程序运行时读取的配置
17
18 客户端标签：
19 [clinet] 所有本地客户端的程序
20 [mysql] mysql命令在本地执行时读取的配置
21 [mysqldump] mysqldump .....

```

第3章 连接管理

1.本地连接

前提条件

```
1 grant all on *.* to test@'localhost' identified by '123';
```

本地账号密码登陆

```
1 mysql -utest -p123
```

本地使用socket登陆

```
1 mysql -utest -p123 -S /tmp/mysql.sock
```

2.远程连接

前提条件

```
1 grant all on *.* to test1@'10.0.0.%' identified by '123';
```

```
1 | mysql -utest1 -p123 -h 10.0.0.51 -P3306
```

3.免交互执行命令

```
1 | mysql -uroot -p123 -e "select user,host from mysql.user;"
```

4.导入数据

```
1 | mysql -uroot -p123 < world.sql
```

5.图形连接

```
1 | sqlyog
2 | navicat
3 | workbench
```

第3章 用户管理

1.用户说明

```
1 | Linux用户:
2 |     登录Linux系统
3 |     管理Linux对象: 文件
4 |
5 | MySQL用户:
6 |     登录MySQL数据库
7 |     管理MySQL对象: 表
```

2.用户的定义

```
1 | Linux用户: 用户名
2 | MySQL用户: 用户名@'白名单'
3 |
4 | 地址列表, 允许白名单的IP登录MySQL, 管理MySQL。
5 | oldya@'localhost' : oldya用户能够通过本地登录MySQL (socket)
6 | oldya@'10.0.0.10' : oldya用户能够通过10.0.0.10远程登录MySQL服务器
7 | oldya@'10.0.0.%' : oldya用户能够通过10.0.0.xx/24远程登录MySQL服务器
8 | oldya@'10.0.0.5%' : ...50-59...
9 | oldya@'10.0.0.0/255.255.254.0'
10 | oldya@'%'
11 | oldya@'db01'
12 | oldya@'db01.oldya.com'
```

3.用户管理

查找用户

```
1 | select user,host,authentication_string from mysql.user;
```

增加用户

```
1 create user oldguo@'localhost';
2 select user,host from mysql.user;
3
4 create user oldboy@'10.0.0.%' identified by '123';
5 select user,host ,authentication_string from mysql.user;
```

root修改用户

```
1 alter user oldguo@'localhost' identified by '123';
```

普通账户修改密码

```
1 set password=password('123456');
```

删除用户

```
1 drop user oldguo@'localhost';
```

第4章 权限管理

1.权限的作用

```
1 用户对数据库对象，有哪些管理能力。
```

2.查看权限列表

```
1 show privileges;
```

3.授权操作

查看帮助:

```
1 help grant
```

格式说明:

```
1 GRANT ALL PRIVILEGES ON *.* TO 'root'@'10.0.0.%' IDENTIFIED BY
  '123456';
2 授权          权限          ON  权限范围  TO  允许用户登陆的主机  创建密码
```

授权举例: 创建远程管理员权限

```
1 GRANT ALL PRIVILEGES ON *,* TO 'root'@'10.0.0.%' IDENTIFIED BY '123456';
```

授权举例: 授权一个普通用户 test@'10.0.0.%',权限为 select、update、delete、insert, 范围: test.*

```
1 | grant select,update,delete,insert on test.* to test@'10.0.0.%' identified by  
   | '123';  
2 | show grants for test@'10.0.0.%';
```

4.回收权限

注意：MySQL中不能重复授权。是相加关系。

```
1 | revoke delete on test.* from 'test'@'10.0.0.%' ;  
2 | show grants for test@'10.0.0.%';
```

5.删除用户

```
1 | DROP USER '用户'@'主机';
```

6.权限表说明

深圳教習
老男仔

表 6-6 MySQL 的 ALL PRIVILEGES 的权限列表

权限	说明
SELECT	查询（数据）
INSERT	插入（数据）
UPDATE	修改（数据）
DELETE	删除（数据）
CREATE	创建（数据库、表等对象）
DROP	删除（数据库、表等对象）
RELOAD	重载
SHUTDOWN	关闭
PROCESS	进程
FILE	文件
REFERENCES	参考资料
INDEX	索引
ALTER	修改（数据库、表等对象）
SHOW DATABASES	查看数据库
SUPER	超级权限
CREATE TEMPORARY TABLES	创建临时表
LOCK TABLES	锁表
EXECUTE	执行
REPLICATION SLAVE	从复制权限
REPLICATION CLIENT	从客户端复制
CREATE VIEW	创建视图
SHOW VIEW	查看视图
CREATE ROUTINE	创建存储过程
ALTER ROUTINE	修改存储过程
CREATE USER	创建用户
EVENT	事件
TRIGGER	触发器
CREATE TABLESPACE	创建表空间

工作中授权时，授权用户应尽量授权为最小的满足业务需求的权限，而不是直接授权为“ALL PRIVILEGES”。

6. 企业中 root 授权权限问题说明

第5章 设置和修改密码

1. 设置root密码

```
1 mysqladmin -uroot password '123456'
2 mysqladmin -uroot password '123456' -s /tmp/mysql_3306.sock
```

2. 修改root密码

方法1: 使用mysqladmin修改


```
1 | mysqladmin -uroot -p123456 password '123'
2 | mysqladmin -uroot -p123456 password '123' -s /tmp/mysql_3306.sock
```

方法2: 使用SQL语句修改指定用户

```
1 | set password for root@localhost =PASSWORD('123');
2 | flush privileges;
```

方法3: 使用SQL语句修改指定用户 5.7的方法

```
1 | UPDATE mysql.user SET authentication_string=PASSWORD("123456") WHERE
   | user='root' and host='localhost';
```

方法4: 使用SQL语句修改当前用户密码

```
1 | set password=password('123456');
2 | flush privileges;
```

第6章 找回root密码

1. 停止MySQL服务

```
1 | systemctl stop mysql
```

2. 忽略授权登陆验证

```
1 | mysqld_safe --skip-grant-tables --user=mysql
```

3. 免密登陆MySQL

```
1 | mysql
```

4. 修改MySQL密码

```
1 | UPDATE mysql.user SET authentication_string=PASSWORD("123456") WHERE
   | user='root' and host='localhost';
```

5. 重启MySQL

```
1 | systemctl restart mysqld
```

6. 使用修改后的密码登陆

```
1 | mysql -uroot -p123456
```

7. 注意

1 | 使用跳过授权表之后,所有的本地和远程用户都不需要用户名密码认证即可连接,非常危险.

深圳教習
徐子豪
老男