MySQL 5.7 安全操作

一、前言

笔者之前写过[《MySQL 性能优化技巧》文章，但没有涉及到 MySQL 安全方面的知识。虽说这是 DBA 需要学习的内容与后端开发人员关系不大，但俗话说技多不压身，即便不深入学习，也需要对其相关内容有所了解。

> 测试环境 MySQL 5.7.20

以下便是笔者浅学后的内容总结。

二、用户相关

创建新用户并合理地设置权限是安全的保障。

2.1 新建用户

```
mysql> create user 用户名 identified by "密码"
```

使用新用户名登录后，由于没有权限只能查看一个数据库：

```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
+--------------------+
1 row in set (0.00 sec)
```

2.2 修改用户名

```
mysql> rename user 旧用户名 to 新用户名
```

2.3 修改密码

不需要登录 MySQL 的情况：

```
shell> mysqladmin -u 用户名 -p password "新密码"

提示需要输入旧密码
```

登陆 MySQL 后，修改新密码的情况，有如下几种方式：

```
mysql> alter user 'root'@'localhost' IDENTIFIED BY '新密码'
```

```
mysql> alter user user() IDENTIFIED BY '新密码'
```

```
mysql> SET PASSWORD FOR 'root'@'localhost' = '新密码'
```

```
mysql> SET PASSWORD = '新密码';
```

```
mysql> grant usage on *.* to "用户名"@"%" identified by "新密码"
```

针对 **MySQL 5.7.6** 以上版本，以上命令在设置密码时，不需要使用 **password()** 给密码加密。

忘记密码，需要重置密码的情况：

1. 关闭 MySQL 服务

```
shell> service mysqld stop
```

1. 创建临时启动文件（/root/mysql-init），内容如下：

```
ALTER USER 'root'@'localhost' IDENTIFIED BY '新密码';
```

1. 启动 MySQL 服务

```
shell> mysqld --user=root --init-file=/root/mysql-init &
```

1. MySQL 服务启动后，删除 /root/mysql-init 文件。

踩坑提醒：

将 /root/mysql-init 文件删除后，笔者执行 service mysqld stop 和 service mysqld start 启动就报错：

```
[ERROR] Could not open unix socket lock file
/var/lib/mysql/mysql.sock.lock
```

通过 ll /var/lib/mysql/ 查看 mysql.sock.lock 所属用户和用户组，发现是 root 权限。

而 mysqld 命令启动使用的是普通用户的权限，因此无法打开文件。

于是笔者将其修改成 mysql 用户和 mysql 组启动成功：

```
chown -R mysql:mysql /var/lib/mysql/mysql.sock

chown -R mysql:mysql /var/lib/mysql/mysql.sock.lock
```

2.4 删除用户

```
mysql> drop user 用户名
```

drop 命令还可以删除多个用户，多个用户名通过“，”隔开。

三、授权相关

MySQL 提供了许多权限类型，读者可以参考文章末尾的资料了解更多知识。

为了测试方便，笔者使用 all 权限进行测试。

3.1 全局级别

授权：

```
mysql> grant all on *.* to 用户名
```

撤销权限：

```
mysql> revoke all on *.* from 用户名
```

其中， * 分别表示库名和表名。

3.2 数据库级别

授权：

```
mysql> grant all on 数据库名.* to 用户名
```

撤销权限：：

```
mysql> revoke all on 数据库名.* from 用户名
```

3.3 表级别

授权：

```
mysql> grant select,insert,update on 数据库名.表名 to 用户名
```

四、备份数据

4.1 物理备份

时机：数据库服务关闭。如果需要运行数据库备份，需要锁定数据库避免在备份期间数据产生变化。

方式：直接拷贝数据库目录和文件（/var/lib/mysql）。

优点：备份速度比逻辑备份快，且包含日志文件和配置文件等信息。

4.2 逻辑备份

时机：数据库服务开启。

4.2.1备份所有数据库

```
shell>mysqldump -h 主机名 -u 用户名 -p --all-databases > dump.sql

提示输入密码
```

如果不是远程备份，主机名参数可以省略。

4.2.2 备份指定数据库

```
shell>mysqldump -h 主机名 -u 用户名 -p --databases 库名1 [库名2 ...] > dump.sql

提示输入密码
```

备份多个数据库，库名之间使用空格隔开。

4.2.3 备份指定数据库表

```
shell>mysqldump -h 主机名 -u 用户名 -p 库名 表名1 [表名2 ...] > dump.sql

提示输入密码
```

备份多张表，表名之间使用空格隔开。

五、还原数据

5.1 物理备份还原

直接将备份目录放在数据库数据目录下（/var/lib/mysql）。

5.2 逻辑备份还原

针对所有数据库：

```
shell> mysql -uroot -p < dump.sql

提示输入密码
```

或

```
mysql> source  dump.sql
```

针对某个库还原：

```
shell> mysql -uroot -p 库名 < dump.sql

提示输入密码
```

六、参考资料

- https://dev.mysql.com/doc/refman/5.7/en/security-against-attack.html MySQL 安全建议
- https://dev.mysql.com/doc/refman/5.7/en/privileges-provided.html 权限类型
- https://dev.mysql.com/doc/refman/5.7/en/backup-types.html 备份相关