

CentOS 下 MySQL 8.0 安装部署，超详细！

👉 mp.weixin.qq.com/s/OQaL0T-jT2xfsJBulau7sQ

点击上方“民工哥技术之路”，选择“设为星标”

回复“1024”获取独家整理的学习资料！



MySQL 8正式版8.0.11已发布，官方表示MySQL8要比MySQL 5.7快2倍，还带来了大量的改进和更快的性能！到底谁最牛呢？请看：[MySQL 5.7 vs 8.0，哪个性能更牛？](#)

MySQL8.0安装 (YUM方式)

1. 首先删除系统默认或之前可能安装的其他版本的mysql

```
[root@DB-node01 ~]# for i in $(rpm -qa|grep mysql);do rpm -e $i --nodeps;done
[root@DB-node01 ~]# rm -rf /var/lib/mysql && rm -rf /etc/my.cnf
```

2. 安装MySQL8.0 的yum资源库

mysql80-community-release-el7-1.noarch.rpm

```
[root@DB-node01 ~]# yum localinstall https://repo.mysql.com//mysql80-community-release-el7-1.noarch.rpm
```

3. 安装MySQL8.0

```
[root@DB-node01 ~]# yum install mysql-community-server
```

#启动MySQL服务器和MySQL的自动启动

```
[root@DB-node01 ~]# systemctl start mysqld
[root@DB-node01 ~]# systemctl enable mysqld
```

4. 使用默认密码初次登录后, 必须要重置密码

查看默认密码, 如下默认密码为"e53xDalx.*dE"

```
[root@DB-node01 ~]# grep 'temporary password' /var/log/mysqld.log
2019-03-06T01:53:19.897262Z 5 [Note] [MY-010454] [Server] A temporary password is generated for root@localhost: e53xDalx.*dE

[root@DB-node01 ~]# mysql -pe53xDalx.*dE
.....
mysql> select version();
ERROR 1820 (HY000): You must reset your password using ALTER USER statement before
```

报错提示必须要重置初始密码, 下面开始重置mysql登录密码 (注意要切换到mysql数据库, 使用use mysql)

```
mysql> use mysql;
ERROR 1820 (HY000): You must reset your password using ALTER USER statement before

mysql> ALTER USER 'root'@'localhost' IDENTIFIED BY '123456';
ERROR 1819 (HY000): Your password does not satisfy the current policy requirements
```

这个其实与validate_password_policy的值有关, mysql8.0更改了validate_password_policy相关的配置名称, 这跟Mysql5.7有点不一样了.

```
mysql> set global validate_password.policy=0;
Query OK, 0 rows affected (0.00 sec)

mysql> set global validate_password.length=1;
Query OK, 0 rows affected (0.00 sec)
```

接着再修改密码

```
mysql> ALTER USER 'root'@'localhost' IDENTIFIED BY '123456';
Query OK, 0 rows affected (0.05 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.03 sec)
```

退出, 重新使用新密码登录mysql

```
[root@DB-node01 ~]# mysql -p123456
.....
mysql> select version();
+-----+
| version() |
+-----+
| 8.0.15    |
+-----+
1 row in set (0.00 sec)
```

查看服务端口

```
mysql> show global variables like 'port';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| port          | 3306  |
+-----+-----+
1 row in set (0.01 sec)
```

查看mysql连接的授权信息

```
mysql> select host,user,password from mysql.user;
ERROR 1054 (42S22): Unknown column 'password' in 'field list'
```

上面这是mysql5.6及以下版本的查看命令,mysql5.7之后的数据库里mysql.user表里已经没有password这个字段了, password字段改成了authentication_string。

```
mysql> select host,user,authentication_string from mysql.user;
+-----+-----+-----+
| host      | user          | authentication_string
+-----+-----+-----+
| localhost | mysql.infoschema | $A$005$THISISACOMBINATIONOFINVALIDSALTANDPASSWORDT
| localhost | mysql.session    | $A$005$THISISACOMBINATIONOFINVALIDSALTANDPASSWORDT
| localhost | mysql.sys        | $A$005$THISISACOMBINATIONOFINVALIDSALTANDPASSWORDT
| localhost | root            | $A$005${7J0=4Dc7Jym8eI/FU4jimKWFvkD9XmoAkF1ca5.Un0
+-----+-----+-----+
4 rows in set (0.00 sec)
```

mysql8.0修改用户密码命令

```
mysql> use mysql;
mysql> ALTER USER 'root'@'localhost' IDENTIFIED BY '123456';
mysql> flush privileges;
```

Mysql8.0安装 (二进制方式)

1. 首先删除系统默认或之前可能安装的其他版本的mysql

```
[root@mysql8-node ~]# for i in $(rpm -qa|grep mysql);do rpm -e $i --nodeps;done
[root@mysql8-node ~]# rm -rf /var/lib/mysql && rm -rf /etc/my.cnf
```

2. 安装需要的软件包

```
[root@mysql8-node ~]# yum -y install libaio
[root@mysql8-node ~]# yum -y install net-tools
```

3. 下载并安装Mysql8.0.12

```
[root@mysql8-node ~]# groupadd mysql
[root@mysql8-node ~]# useradd -g mysql mysql
```

```
[root@mysql8-node ~]# cd /usr/local/src/
[root@mysql-node src]# ll
-rw-r--r-- 1 root root 620389228 Aug 22 2018 mysql8.0.12_bin_centos7.tar.gz
[root@mysql-node src]# tar -zxvf mysql8.0.12_bin_centos7.tar.gz
[root@mysql-node src]# mv mysql /usr/local/
[root@mysql-node src]# chown -R mysql.mysql /usr/local/mysql

[root@mysql-node src]# vim /home/mysql/.bash_profile
export PATH=/usr/local/mysql/bin:/usr/local/mysql/lib:$PATH
[root@mysql-node src]# source /home/mysql/.bash_profile
[root@mysql-node src]# echo "PATH=$PATH:/usr/local/mysql/bin" >> /etc/profile
[root@mysql-node src]# source /etc/profile
```

4. 创建数据目录

```
[root@mysql-node src]# mkdir -p /data/mysql/{data,log,binlog,conf,tmp}
[root@mysql-node src]# chown -R mysql.mysql /data/mysql
```

5. 配置mysql

```

[root@mysql-node src]# su - mysql
[mysql@mysql-node ~]$ vim /data/mysql/conf/my.cnf
[mysqld]
lower_case_table_names      = 1
user                        = mysql
server_id                   = 1
port                        = 3306


default-time-zone = '+08:00'
enforce_gtid_consistency   = ON
gtid_mode                 = ON
binlog_checksum            = none
default_authentication_plugin = mysql_native_password
datadir                    = /data/mysql/data
pid-file                   = /data/mysql/tmp/mysqld.pid
socket                     = /data/mysql/tmp/mysqld.sock
tmpdir                     = /data/mysql/tmp/
skip-name-resolve          = ON
open_files_limit           = 65535
table_open_cache           = 2000


#####innodb#####
innodb_data_home_dir        = /data/mysql/data
innodb_data_file_path       = ibdata1:512M;ibdata2:512M:autoextend
innodb_buffer_pool_size    = 12000M
innodb_flush_log_at_trx_commit = 1
innodb_io_capacity         = 600
innodb_lock_wait_timeout   = 120
innodb_log_buffer_size     = 8M
innodb_log_file_size       = 200M
innodb_log_files_in_group  = 3
innodb_max_dirty_pages_pct = 85
innodb_read_io_threads     = 8
innodb_write_io_threads    = 8
innodb_thread_concurrency  = 32
innodb_file_per_table
innodb_rollback_on_timeout


innodb_undo_directory      = /data/mysql/data
innodb_log_group_home_dir  = /data/mysql/data


#####session#####
join_buffer_size = 8M
key_buffer_size = 256M
bulk_insert_buffer_size = 8M
max_heap_table_size = 96M
tmp_table_size = 96M
read_buffer_size = 8M
sort_buffer_size = 2M
max_allowed_packet = 64M
read_rnd_buffer_size = 32M

```

```
#####log set#####
log-error                = /data/mysql/log/mysql.err
log-bin                  = /data/mysql/binlog/binlog
log_bin_index            = /data/mysql/binlog/binlog.index
max_binlog_size          = 500M
slow_query_log_file      = /data/mysql/log/slow.log
slow_query_log           = 1
long_query_time          = 10
log_queries_not_using_indexes = ON
log_throttle_queries_not_using_indexes = 10
log_slow_admin_statements = ON
log_output               = FILE, TABLE
master_info_file         = /data/mysql/binlog/master.info
```

6. 初始化 (稍等一会儿, 可以到/data/mysql/log/mysql.err日子里查看初始化过程, 看看有没有error信息)

```
[mysql@mysql-node ~]$ mysqld --defaults-file=/data/mysql/conf/my.cnf --
initialize-insecure --user=mysql
```

7. 启动mysqld

```
[mysql@mysql-node ~]$ mysqld_safe --defaults-file=/data/mysql/conf/my.cnf &
[mysql@mysql-node ~]$ lsof -i:3306
COMMAND  PID  USER  FD  TYPE  DEVICE SIZE/OFF NODE NAME
mysqld   24743  mysql  23u  IPv6  23132988      0t0  TCP *:mysql (LISTEN)
```

8. 登录mysql, 重置密码

本地首次使用sock文件登录mysql是不需要密码的

```
[mysql@mysql-node ~]# mysql -S /data/mysql/tmp/mysqld.sock
```

```
.....
mysql> ALTER USER 'root'@'localhost' IDENTIFIED BY '123456';
Query OK, 0 rows affected (0.07 sec)
```

```
mysql> flush privileges;
Query OK, 0 rows affected (0.03 sec)
```

```
mysql> select host,user,authentication_string from mysql.user;
+-----+-----+-----+
| host      | user                | authentication_string |
+-----+-----+-----+
| localhost | mysql.infoschema    | $A$005$THISISACOMBINATIONOFINVALIDSALTANDPASSWORDT |
| localhost | mysql.session       | $A$005$THISISACOMBINATIONOFINVALIDSALTANDPASSWORDT |
| localhost | mysql.sys           | $A$005$THISISACOMBINATIONOFINVALIDSALTANDPASSWORDT |
| localhost | root                | *6BB4837EB74329105EE4568DDA7DC67ED2CA2AD9          |
+-----+-----+-----+
4 rows in set (0.00 sec)
```

退出, 此时密码重置后, 就不能使用sock文件无密码登录了

```
[root@mysql-node ~]# mysql -S /data/mysql/tmp/mysql.sock
ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: NO)

[root@mysql-node ~]# mysql -p123456
mysql: [Warning] Using a password on the command line interface can be insecure.
ERROR 2002 (HY000): Can't connect to local MySQL server through socket '/tmp/mysql.
```

做sock文件的软链接

```
[root@mysql-node ~]# ln -s /data/mysql/tmp/mysql.sock /tmp/mysql.sock
```

登录

```
[root@mysql-node ~]# mysql -p123456
```

或者

```
[root@mysql-node ~]# mysql -uroot -S /data/mysql/tmp/mysql.sock -p123456
```

```
mysql> select version();
```

```
+-----+
```

```
| version() |
```

```
+-----+
```

```
| 8.0.12    |
```

```
+-----+
```

```
1 row inset (0.00 sec)
```

#授予用户权限。必须先要创建用户，才能授权!!

(创建用户时要带@并指定地址，则grant授权时的地址就是这个@后面指定的!，否则grant授权就会报错!)

```
mysql> create user 'kevin'@'%' identified by '123456';
```

```
Query OK, 0 rows affected (0.11 sec)
```

```
mysql> grant all privileges on *.* to 'kevin'@'%' with grant option;
```

```
Query OK, 0 rows affected (0.21 sec)
```

```
mysql> select host,user,authentication_string from mysql.user;
```

```
+-----+-----+-----+
```

```
-----+
```

```
| host      | user          | authentication_string
```

```
+-----+-----+-----+
```

```
-----+
```

```
| %         | kevin         | *6BB4837EB74329105EE4568DDA7DC67ED2CA2AD9
```

```
| localhost | mysql.infoschema | $A$005$THISISACOMBINATIONOFINVALIDSALTANDPASSWORDT
```

```
| localhost | mysql.session   | $A$005$THISISACOMBINATIONOFINVALIDSALTANDPASSWORDT
```

```
| localhost | mysql.sys       | $A$005$THISISACOMBINATIONOFINVALIDSALTANDPASSWORDT
```

```
| localhost | root           | *6BB4837EB74329105EE4568DDA7DC67ED2CA2AD9
```

```
+-----+-----+-----+
```

```
-----+
```

```
5 rows inset (0.00 sec)
```

```
mysql> update mysql.user set host='172.16.60.%' where user="kevin";
```

```
Query OK, 1 row affected (0.16 sec)
```

```
Rows matched: 1  Changed: 1  Warnings: 0
```

```
mysql> flush privileges;
```

```
Query OK, 0 rows affected (0.05 sec)
```

```
mysql> select host,user,authentication_string from mysql.user;
```

```
+-----+-----+-----+
```

```
-----+
```

```
| host      | user          | authentication_string
```

```
+-----+-----+-----+
```

```
-----+
```

```
| 172.16.60.% | kevin         | *6BB4837EB74329105EE4568DDA7DC67ED2CA2AD9
```



```

| localhost | mysql.infoschema | $A$005$THISISACOMBINATIONOFINVALIDSALTANDPASSWOR
| localhost | mysql.session    | $A$005$THISISACOMBINATIONOFINVALIDSALTANDPASSWOR
| localhost | mysql.sys        | $A$005$THISISACOMBINATIONOFINVALIDSALTANDPASSWOR
| localhost | root             | *6BB4837EB74329105EE4568DDA7DC67ED2CA2AD9
+-----+-----+-----+
-----+
5 rows inset (0.00 sec)

```

```

mysql> create user 'bobo'@'172.16.60.%' identified by '123456';
Query OK, 0 rows affected (0.09 sec)

```

```

mysql> grant all privileges on *.* to 'bobo'@'172.16.60.%;
Query OK, 0 rows affected (0.17 sec)

```

```

mysql> flush privileges;
Query OK, 0 rows affected (0.04 sec)

```

```

mysql> select host,user,authentication_string from mysql.user;
+-----+-----+-----+
-----+
| host      | user              | authentication_string
+-----+-----+-----+
-----+
| 172.16.60.% | bobo              | *6BB4837EB74329105EE4568DDA7DC67ED2CA2AD9
| 172.16.60.% | kevin             | *6BB4837EB74329105EE4568DDA7DC67ED2CA2AD9
| localhost   | mysql.infoschema | $A$005$THISISACOMBINATIONOFINVALIDSALTANDPASSWOR
| localhost   | mysql.session    | $A$005$THISISACOMBINATIONOFINVALIDSALTANDPASSWOR
| localhost   | mysql.sys        | $A$005$THISISACOMBINATIONOFINVALIDSALTANDPASSWOR
| localhost   | root             | *6BB4837EB74329105EE4568DDA7DC67ED2CA2AD9
+-----+-----+-----+
-----+
6 rows inset (0.00 sec)

```

```

mysql> show grants for kevin@'172.16.60.%;
+-----+
-----+
-----+
-----+
-----+
-----+
| Grants for kevin@172.16.60.%
+-----+
-----+
-----+
-----+
-----+
-----+
| GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, RELOAD, SHUTDOWN, PROCESS, FI
+-----+
-----+
-----+
-----+
-----+
1 row in set (0.00 sec)

```

MySQL单机多实例安装配置

通过上面二进制部署可知, 已经起来一个3306端口的MySQL实例, 现在需要再起来两个实例, 分别为3307, 3308. 操作如下:

创建实例的数据目录

```
[root@mysql-node ~]# mkdir -p /data/mysql3307/{data,log,binlog,conf,tmp}
[root@mysql-node ~]# mkdir -p /data/mysql3308/{data,log,binlog,conf,tmp}
[root@mysql-node ~]# chown -R mysql:mysql /data/mysql3307
[root@mysql-node ~]# chown -R mysql:mysql /data/mysql3308
```

配置mysql

```
[root@mysql-node ~]# cp -r /data/mysql/conf/my.cnf /data/mysql3307/conf/
[root@mysql-node ~]# cp -r /data/mysql/conf/my.cnf /data/mysql3308/conf/
[root@mysql-node ~]# sed -
i 's#/data/mysql/#/data/mysql3307/#g' /data/mysql3307/conf/my.cnf
[root@mysql-node ~]# sed -
i 's#/data/mysql/#/data/mysql3308/#g' /data/mysql3308/conf/my.cnf
[root@mysql-node ~]# sed -i 's/3306/3307/g' /data/mysql3307/conf/my.cnf
[root@mysql-node ~]# sed -i 's/3306/3308/g' /data/mysql3308/conf/my.cnf
[root@mysql-node ~]# chown -R mysql:mysql /data/mysql*
```

进行初始化两个实例

```
[root@mysql-node ~]# mysqld --defaults-file=/data/mysql3307/conf/my.cnf --
initialize-insecure --user=mysql
[root@mysql-node ~]# mysqld --defaults-file=/data/mysql3308/conf/my.cnf --
initialize-insecure --user=mysql
```

接着启动mysqld

```
[root@mysql-node ~]# mysqld_safe --defaults-file=/data/mysql3307/conf/my.cnf &
[root@mysql-node ~]# mysqld_safe --defaults-file=/data/mysql3308/conf/my.cnf &
```

查看启动是否成功

```
[root@mysql-node ~]# ps -ef|grep mysql
mysql      23996      1  0 14:37 ?          00:00:00 /bin/sh /usr/local/mysql/bin/mysqld
--defaults-file=/data/mysql/conf/my.cnf
mysql      24743 23996  0 14:38 ?          00:00:17 /usr/local/mysql/bin/mysqld --
defaults-file=/data/mysql/conf/my.cnf --basedir=/usr/local/mysql --
datadir=/data/mysql/data --plugin-dir=/usr/local/mysql/lib/plugin --log-
error=/data/mysql/log/mysqld.err --open-files-limit=65535 --pid-
file=/data/mysql/tmp/mysqld.pid --socket=/data/mysql/tmp/mysqld.sock --port=3306
root       30473 23727  0 15:33 pts/0    00:00:00 /bin/sh /usr/local/mysql/bin/mysqld
--defaults-file=/data/mysql3307/conf/my.cnf
mysql      31191 30473 17 15:33 pts/0    00:00:02 /usr/local/mysql/bin/mysqld --
defaults-file=/data/mysql3307/conf/my.cnf --basedir=/usr/local/mysql --
datadir=/data/mysql3307/data --plugin-dir=/usr/local/mysql/lib/plugin --
user=mysql --log-error=/data/mysql3307/log/mysqld.err --open-files-limit=65535 --
pid-file=/data/mysql3307/tmp/mysqld.pid --socket=/data/mysql3307/tmp/mysqld.sock -
-port=3307
root       31254 23727  0 15:33 pts/0    00:00:00 /bin/sh /usr/local/mysql/bin/mysqld
--defaults-file=/data/mysql3308/conf/my.cnf
mysql      31977 31254 39 15:33 pts/0    00:00:02 /usr/local/mysql/bin/mysqld --
defaults-file=/data/mysql3308/conf/my.cnf --basedir=/usr/local/mysql --
datadir=/data/mysql3308/data --plugin-dir=/usr/local/mysql/lib/plugin --
user=mysql --log-error=/data/mysql3308/log/mysqld.err --open-files-limit=65535 --
pid-file=/data/mysql3308/tmp/mysqld.pid --socket=/data/mysql3308/tmp/mysqld.sock -
-port=3308
root       32044 23727  0 15:34 pts/0    00:00:00 grep --color=auto mysql
```

```
[root@mysql-node ~]# lsof -i:3307
COMMAND  PID  USER  FD  TYPE  DEVICE  SIZE/OFF  NODE NAME
mysqld   31191 mysql  22u  IPv6  23144844      0t0  TCP *:opsession-prxy (LISTEN)
[root@mysql-node ~]# lsof -i:3308
COMMAND  PID  USER  FD  TYPE  DEVICE  SIZE/OFF  NODE NAME
mysqld   31977 mysql  22u  IPv6  23145727      0t0  TCP *:tns-server (LISTEN)
[root@mysql-node ~]# lsof -i:3306
COMMAND  PID  USER  FD  TYPE  DEVICE  SIZE/OFF  NODE NAME
mysqld   24743 mysql  23u  IPv6  23132988      0t0  TCP *:mysql (LISTEN)
```

登录3307端口实例, 并设置密码

```
[root@mysql-node ~]# mysql -S /data/mysql3307/tmp/mysqld.sock
.....
mysql> ALTER USER 'root'@'localhost' IDENTIFIED BY '123456';
Query OK, 0 rows affected (0.11 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.11 sec)
```

退出, 使用新密码登录

```
[root@mysql-node ~]# mysql -uroot -S /data/mysql3307/tmp/mysqld.sock -p123456
.....
mysql>
```

同理, 登录3308端口实例, 并设置密码

```
[root@mysql-node ~]# mysql -S /data/mysql3308/tmp/mysql.sock
.....
mysql> ALTER USER 'root'@'localhost' IDENTIFIED BY '123456';
Query OK, 0 rows affected (0.13 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.03 sec)
```

退出, 使用新密码登录

```
[root@mysql-node ~]# mysql -uroot -S /data/mysql3308/tmp/mysql.sock -p123456
.....
mysql>
```

3306, 3307, 3308三个端口实例的启动命令分别为:

```
mysqld_safe --defaults-file=/data/mysql/conf/my.cnf &
mysqld_safe --defaults-file=/data/mysql3307/conf/my.cnf &
mysqld_safe --defaults-file=/data/mysql3308/conf/my.cnf &
```

登录命令分别为:

```
mysql -uroot -S /data/mysql/tmp/mysql.sock -p123456
mysql -uroot -S /data/mysql3307/tmp/mysql.sock -p123456
mysql -uroot -S /data/mysql3308/tmp/mysql.sock -p123456
```

不过为了解决大家平时重复安装的问题, 特意将多实例安装方法编辑成脚本了, 有需要的读者可以在本公众号后台直接回复 **MySQL8** 获取多实例安装脚本。

MySQL8.0使用过程中踩过的一些坑

1)创建用户和授权 在mysql8.0创建用户和授权和之前不太一样了, 其实严格上来讲, 也不能说是不一样, 只能说是更严格, mysql8.0需要先创建用户(创建用户时要带@并指定地址, 则grant授权时的地址就是这个@后面指定的!, 否则grant授权就会报错!)和设置密码, 然后才能授权。

```
mysql> create user 'kevin'@'%' identified by '123456';
Query OK, 0 rows affected (0.04 sec)
```

```
mysql> grant all privileges on *.* to 'kevin'@'%' with grant option;
Query OK, 0 rows affected (0.04 sec)
```

```
mysql> create user 'bobo'@'%' identified by '123456';
Query OK, 0 rows affected (0.06 sec)
```

```
mysql> grant all privileges on *.* to 'bobo'@'%' with grant option;
Query OK, 0 rows affected (0.03 sec)
```

```
mysql> flush privileges;
Query OK, 0 rows affected (0.04 sec)
```

```
mysql> select host,user,authentication_string from mysql.user;
+-----+-----+-----+
| host      | user                | authentication_string |
+-----+-----+-----+
| %         | bobo                | $A$005$1VY")q?G6<^X@-6LsXrPt5C0Tw1TuvHba0a3sYF0DKViIGoRPuCF8AzwiFcim1 |
| %         | kevin               | $A$005$hy`U}ZB#R::rA8W0y2rmwgySqzv0rmR1eTeNDSaXfQP |
| localhost | mysql.infoschema    | $A$005$THISISACOMBINATIONOFINVALIDSALTANDPASSWORDT |
| localhost | mysql.session       | $A$005$THISISACOMBINATIONOFINVALIDSALTANDPASSWORDT |
| localhost | mysql.sys           | $A$005$THISISACOMBINATIONOFINVALIDSALTANDPASSWORDT |
| localhost | root                | $A$005$/VO_y^7,]6;2qxggBLmJzhA0Qylu5/AHuRScZ/ykKed |
+-----+-----+-----+
```

如果还是用Mysql5.7及之前版本的直接授权的方法,会有报错:

```
mysql> grant all privileges on *.* to 'shibo'@'%' identified by '123456';
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that cor
```

2)Mysql8.0默认是不能使用root账号进行远程登录的! root账号只能本地登录!

```
mysql> select host,user,authentication_string from mysql.user;
+-----+-----+-----+
| host      | user                | authentication_string |
+-----+-----+-----+
| %         | bobo                | $A$005$1VY")q?      |
G6<^X@-6LsXrPt5C0Tw1TuvHba0a3sYF0DKViIGoRPuCF8AzwiFcim1 |
| %         | kevin               | $A$005$hy`U}ZB#R::rA8W0y2rmwgySqzv0rmR1eTeNDSaXfQP |
| localhost | mysql.infoschema    | $A$005$THISISACOMBINATIONOFINVALIDSALTANDPASSWORDT |
| localhost | mysql.session       | $A$005$THISISACOMBINATIONOFINVALIDSALTANDPASSWORDT |
| localhost | mysql.sys           | $A$005$THISISACOMBINATIONOFINVALIDSALTANDPASSWORDT |
| localhost | root                | $A$005$/VO_y^7,]6;2qxggBLmJzhA0Qylu5/AHuRScZ/ykKed |
+-----+-----+-----+
6 rows in set (0.00 sec)
```

如果想要远程登录,则需要进行update更新下root账号的权限

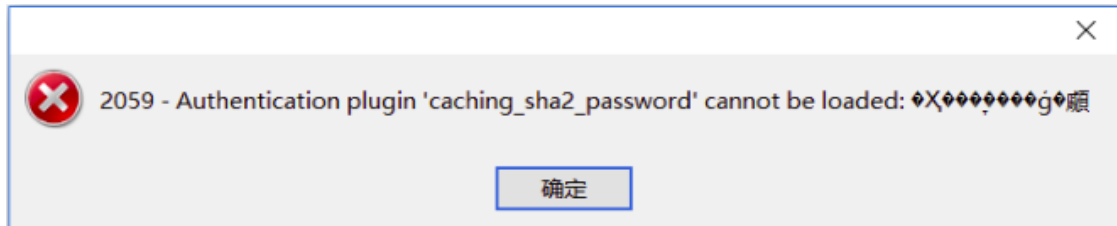
```
mysql> update mysql.user set host='% 'where user="root";
Query OK, 1 row affected (0.10 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

```
mysql> flush privileges;
Query OK, 0 rows affected (0.14 sec)
```

```
mysql> select host,user,authentication_string from mysql.user;
+-----+-----+-----+
| host      | user                | authentication_string |
+-----+-----+-----+
| %         | bobo                | $A$005$1VY")q?      |
G6<^X@-6LsXrPt5C0Tw1TuvHba0a3sYF0DKViIGoRPuCF8AzwiFcim1 |
| %         | kevin               | $A$005$hy`U}ZB#R::rA8W0y2rmwgySqzv0rmR1eTeNDSaXfQP |
| %         | root                | $A$005$/VO_y^7,]6;2qxggBLmJzhA0Qylu5/AHuRScZ/ykKed |
| localhost | mysql.infoschema    | $A$005$THISISACOMBINATIONOFINVALIDSALTANDPASSWORDT |
| localhost | mysql.session       | $A$005$THISISACOMBINATIONOFINVALIDSALTANDPASSWORDT |
| localhost | mysql.sys           | $A$005$THISISACOMBINATIONOFINVALIDSALTANDPASSWORDT |
+-----+-----+-----+
6 rows in set (0.00 sec)
```

这样就能在远程使用root账号登录该mysql8.0的数据库了

修改root账号权限, 允许root账号远程登录后, 用navicat进行mysql的远程连接时, 出现了弹窗报错:



出现这个原因是mysql8 之前的版本中加密规则是mysql_native_password, 而在mysql8之后,加密规则是caching_sha2_password, 解决问题方法有两种:
一种是升级navicat驱动;
一种是把mysql用户登录密码加密规则还原成mysql_native_password; 这里选择第二种方法来解决:

#修改加密规则

```
mysql> ALTER USER 'root'@'%' IDENTIFIED BY '123456' PASSWORD EXPIRE NEVER;  
Query OK, 0 rows affected (0.16 sec)
```

#更新一下用户的密码

```
mysql> ALTER USER 'root'@'%' IDENTIFIED WITH mysql_native_password BY '123456';  
Query OK, 0 rows affected (0.08 sec)
```

#刷新权限

```
mysql> FLUSH PRIVILEGES;  
Query OK, 0 rows affected (0.03 sec)
```

这样问题就解决了。

- 1、使用sqlyog链接时会出现2058的异常, 此时我们需要修改mysql, 命令行登录mysql (与修改密码中登录相同, 使用修改后的密码), 然后执行下面的命令: `mysql > ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'password';` 其中password为自己修改的密码。然后SQLyog中重新连接, 则可连接成功, OK。
- 2、如果报错: `ERROR 1396 (HY000): Operation ALTER USER failed for 'root'@'localhost'`则使用下面命令: `mysql > ALTER USER 'root'@'%' IDENTIFIED WITH mysql_native_password BY 'password';`
sqlyog链接时出现2058异常
修改默认编码方式 mysql8.0默认编码方式为utf8mb4, 因此使用时不需要修改, 可使用如下命令查看:

```
mysql > SHOW VARIABLES WHERE Variable_name LIKE 'character_set_%' OR Variable_name
```

如果需要修改其他编码方式, 比如需要修改为utf8mb4, 可以使用如下方式:

修改mysql配置文件my.cnf，找到后请在以下三部分里添加如下内容：

```
[client]
default-character-set = utf8mb4
[mysql]
default-character-set = utf8mb4
[mysqld]
character-set-client-handshake = FALSE
character-set-server = utf8mb4
collation-server = utf8mb4_unicode_ci
init_connect='SET NAMES utf8mb4'
```

然后重启mysqld服务即可, 其中:

```
character_set_client  (客户端来源数据使用的字符集)
character_set_connection (连接层字符集)
character_set_database (当前选中数据库的默认字符集)
character_set_results (查询结果字符集)
character_set_server (默认的内部操作字符集)
```

数据库连接参数中:

characterEncoding=utf8 会被自动识别为utf8mb4，也可以不加这个参数，会自动检测。
而autoReconnect=true 是必须加上的。

6)部分参数配置查询命令


```
#查询mysql最大连接数设置
mysql> show global variables like 'max_conn%';
mysql> SELECT @@MAX_CONNECTIONS AS 'Max Connections';
```

```
# 查看最大链接数
mysql> show global status like 'Max_used_connections';
```

```
# 查看慢查询日志是否开启以及日志位置
mysql> show variables like 'slow_query%';
```

```
# 查看慢查询日志超时记录时间
mysql> show variables like 'long_query_time';
```

```
# 查看链接创建以及现在正在链接数
mysql> show status like 'Threads%';
```

```
# 查看数据库当前链接
mysql> show processlist;
```

```
# 查看数据库配置
mysql> show variables like '%quer%';
```

作者：散尽浮华

出处：<https://www.cnblogs.com/kevingrace/p/10482469.html>





[推荐阅读点击标题可跳转](#)

[CentOS 搭建 K8S，一次性成功，收藏了！](#)

[可怕！公司部署了一个东西，悄悄盯着你...](#)

[牛逼！白帽子们把色情版"微信"的底裤扒了个底朝天.....](#)

[请收藏！Linux 运维必备的 40 个命令总结](#)

[超全的数据库建表/SQL/索引规范，适合贴在工位上！](#)

[MySQL 5.7 vs 8.0，哪个性能更牛？](#)

[推荐 11 个极易上手的 Docker 实践项目](#)

[13 款 Linux 实用工具推荐，个个是神器！](#)

