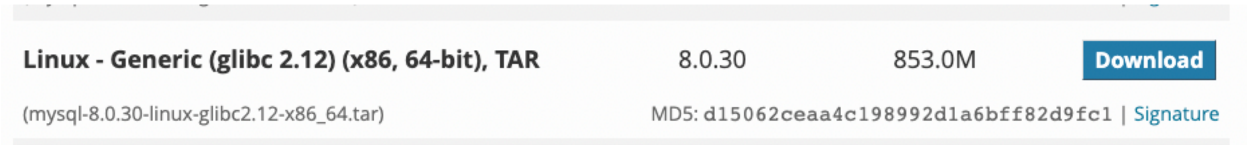
一、官网下载Greate包：

官网地址：https://dev.mysql.com/downloads/mysql/



Linux - Generic (glibc 2.12) (x86, 64-bit), TAR 8.0.30 853.0M

(mysql-8.0.30-linux-glibc2.12-x86_64.tar) MD5: d15062ceaa4c198992d1a6bff82d9fc1 | Signature

二、配置初始化环境：

```
一、依赖包 yum 源安装：
1、二进制包环境：
yum -y install  wget make gcc gcc-c++ autoconf automake zlib* libxml2* ncurs

2、源码包环境：
yum -y install devtoolset-10-gcc devtoolset-10-gcc-c++ devtoolset- 10-binuti
yum install centos-release-scl -y
scl enable devtoolset-10 bash
echo "source /opt/rh/devtoolset-10/enable" >>/etc/profile  // 自定义修改路径
// 查看 yum 包：yum search aio



二、初始化系统环境：
（一）关闭 numa 四种方式：
// 在bios层面numa关闭时，无论os层面的numa是否打开，都不会影响性能。
// 一定在新服务器上线前修改，后期修改需要重启服务器。

1、 bios 级别：（建议使用，如上图）
[root@dongye-es-test-3 ~]#  yum install numactl  -y
[root@dongye-es-test-3 ~]# numactl --hardware|grep available
available: 1 nodes (0) #如果是2或多个nodes就说明numa没关掉

2. OS grub级别：
vi /boot/grub2/grub.cfg
#/* Copyright 2010, Oracle. All rights reserved. */
default=0
timeout=5
hiddenmenu
foreground=000000
background=ffffff splashimage=(hd0,0)/boot/grub/oracle.xpm.gz
```

```
title Trying_C0D0_as_HD0
root (hd0,0)
kernel /boot/vmlinuz-2.6.18-128.1.16.0.1.el5 root=LABEL=DBSYS ro bootarea=db
initrd /boot/initrd-2.6.18-128.1.16.0.1.el5.img

// 在os层numa关闭时,打开bios层的numa会影响性能，QPS会下降15-30%;


3、数据库级别：
mysql> show variables like '%numa%';
+-------------------------+-------+
| Variable_name           | Value |
+-------------------------+-------+
| innodb_numa_interleave | OFF   |
+-------------------------+-------+

4、启动文件中关闭：
vi /etc/init.d/mysqld
// 找到如下行
# Give extra arguments to mysqld with the my.cnf file. This script # may be
$bindir/mysqld_safe --datadir="$datadir" --pid-file="$mysqld_pid_file_path"
wait_for_pid created "$!" "$mysqld_pid_file_path"; return_value=$?

// 将$bindir/mysqld_safe --datadir="$datadir" 这一行修改为:
/usr/bin/numactl --interleave all $bindir/mysqld_safe -- datadir="$datadir"
wait_for_pid created "$!" "$mysqld_pid_file_path"; return_value=$?



（二）、启用io调度器：
// SAS : deadline
// SSD&PCI-E: noop
// 注意：centos 7 默认是deadline

1、永久：vi /boot/grub2/grub.cfg
kernel /boot/vmlinuz-2.6.18-8.el5 ro root=LABEL=/ elevator=deadline rhgb qui

2、临时 修改为 deadline（centos6）
cd  /sys/block/磁盘标识/queue
echo "deadline" > /sys/block/磁盘标识/queue/scheduler

3、查看IO调度器：
dmesg |grep -i scheduler
```

```



（三）、关闭大页内存：
1、vim /etc/rc.local 在文件末尾添加如下指令：
if test -f /sys/kernel/mm/transparent_hugepage/defrag; then
    echo never > /sys/kernel/mm/transparent_hugepage/defrag
fi
if test -f /sys/kernel/mm/transparent_hugepage/enabled; then
    echo never > /sys/kernel/mm/transparent_hugepage/enabled
fi

2、授权启动文件：
[root@localhost ~]# chmod +x /etc/rc.d/rc.local

3、生效启动文件配置
[root@localhost ~]# source /etc/rc.local

4、查看设置：
[root@master ~]# cat /sys/kernel/mm/transparent_hugepage/enabled
always madvise [never]

[root@master ~]# cat /sys/kernel/mm/transparent_hugepage/defrag
always madvise [never]



（四）、调整文件句柄和用户进程数量：
1、vim /etc/sysctl.conf
vm.swappiness = 5                                  # 注释：内存剩余 5% 的时候
vm.dirty_ratio = 20                                # 注释：百分比值。当脏数据
vm.dirty_background_ratio = 10                     # 注释：同上，默认值为10。
net.ipv4.tcp_max_syn_backlog = 819200
net.core.netdev_max_backlog = 400000
net.core.somaxconn = 4096
net.ipv4.tcp_tw_reuse=1
net.ipv4.tcp_tw_recycle=0

[root@maxscale-test ~]# sysctl -p

2、vi /etc/security/limits.conf
# 文件产生时，同样会生成文件句柄（文件存储单元与文件标识符的映射链），os会通过文件句
root soft nofile 65535
root hard nofile 65535
* soft nofile 65535
* hard nofile 65535



（五）、文件系统的优化：
1、安装xfs
yum install xfsprogs kmod-xfs xfsdump xfsprogs-devel
mkfs.xfs /dev/vdb
partprobe

2、修改/etc/fstab
vim /etc/fstab
#/dev/vdb    /data   xfs  defaults,noatime,nodiratime,nobarrier,nodelalloc
/dev/vdb    /data   xfs  defaults,noatime,nodiratime,nobarrier 0 0


3、挂载xfs磁盘到/data
mkdir /data
mount -t xfs -o defaults,noatime,nodiratime,nobarrier /dev/vdb /data
# mount -t ext4 -o defaults,noatime,nodiratime,nobarrier,nodelalloc /dev/sdd



（六）、调整 mysqld 的优先级最高：（防止系统OOM掉mysqld进程）
renice -19 -p `pidof mysqld`;renice -19 -p `pidof mysqld_safe`



（七）、防火墙：
禁用selinux : vim /etc/sysconfig/selinux 更改SELINUX=disabled。
setenforce 0


iptables:  如果不使用可以关闭。如果使用则需要打开MySQL需要的端口号。
systemctl stop firewalld
systemctl disable firewalld


（八）、重启服务器：
reboot

```

```


（九）、检查io调度器、numa、xfs配置是否生效：
mount
dmesg |grep -i scheduler
grep -i numa /var/log/dmesg
```

三、 MySQL 运行环境初始化:

```
# 下载 MySQL 安装包：
wget https://dev.mysql.com/get/downloads/mysql/mysql-8.4.5-linux-glibc2.28-x8
tar -xvf mysql-8.4.5-linux-glibc2.28-x86_64.tar.xz

wget https://downloads.percona.com/downloads/percona-toolkit/3.7.0/binary/tar
wget https://downloads.percona.com/downloads/percona-toolkit/3.7.0/binary/tar

1、创建MySQL用户
groupadd -g 2023 mysql;useradd -g mysql -u 2023  -s /sbin/nologin -d /usr/loc

注释：
// -g 手工指定用户的初始组。一般以和用户名相同的组作为用户的初始组，在创建用户时会默
// -s 手工指定用户的登录 Shell，默认是 /bin/bash；
// -d 手工指定用户的主目录。主目录必须写绝对路径，而且如果需要手工指定主目录，则一定
// -M 不建立使用者目录，即使/etc/login.defs系统档设定要建立使用者目录。
// -N no-user-group不创建与用户同名的组



2、解压软件到指定目录：
mkdir -p /data/db_soft/mysql/oracle/mysql8030/
tar -xvf  mysql-8.0.30-linux-glibc2.12-x86_64.tar -C  /data/db_soft/mysql/ora
cd /data/db_soft/mysql/oracle/mysql8030/
tar -xf mysql-8.0.30-linux-glibc2.12-x86_64.tar.xz



3、创建软连接到 /usr/local/mysql 目录下
ln -s /data/db_soft/mysql/oracle/mysql8030/mysql-8.0.30-linux-glibc2.12-x86_6
chown mysql:mysql  /usr/local/mysql8030 -R
chown mysql:mysql /data/db_soft/mysql/oracle/mysql8030 -R
ll -d   /usr/local/mysql8030
```

```


4、安装 PXB 备份软件
mkdir -p /usr/local/percona-xtrabackup-8.0.30
tar -xf /data/db_soft/mysql/tools/PXB/percona-xtrabackup-8.0.30-23-Linux-x86_
ln -s /data/db_soft/mysql/tools/PXB/percona-xtrabackup-8.0.30-23-Linux-x86_64
chown mysql:mysql  /data/db_soft/mysql/tools/PXB/percona-xtrabackup-8.0.30-23
chown mysql:mysql  /usr/local/percona-xtrabackup-8.0.30 -R



5、percona-toolkit 工具安装：
mkdir /usr/local/percona-toolkit
tar -xf /data/db_soft/mysql/tools/percona-toolkit-3.5.1_x86_64.tar.gz /data/c
ln -s /data/db_soft/mysql/tools/percona-toolkit-3.5.1/bin /usr/local/percona-
chown  -R mysql:mysql /data/db_soft/mysql/tools/percona-toolkit-3.5.1
chown -R mysql:mysql /usr/local/percona-toolkit



6、设置环境变量：
echo "export PATH=/usr/local/mysql8030/bin:/usr/local/percona-xtrabackup-8.0.
which mysql



7、检查mysqld的环境依赖：
[root@dongye-es-test-1 mysql8030]# ldd /usr/local/mysql8030/bin/mysqld
        linux-vdso.so.1 =>  (0x00007ffd1edf3000)
        libpthread.so.0 => /lib64/libpthread.so.0 (0x00007f0393d67000)
        libdl.so.2 => /lib64/libdl.so.2 (0x00007f0393b63000)
        librt.so.1 => /lib64/librt.so.1 (0x00007f039395b000)
        libcrypto.so.1.1 => /usr/local/mysql8030/bin/../lib/private/libcrypto
        libssl.so.1.1 => /usr/local/mysql8030/bin/../lib/private/libssl.so.1.
        libprotobuf-lite.so.3.19.4 => /usr/local/mysql8030/bin/../lib/private
        libaio.so.1 => /lib64/libaio.so.1 (0x00007f0392d73000)
        libnuma.so.1 => /lib64/libnuma.so.1 (0x00007f0392b67000)
        libstdc++.so.6 => /lib64/libstdc++.so.6 (0x00007f039285f000)
        libm.so.6 => /lib64/libm.so.6 (0x00007f039255d000)
        libgcc_s.so.1 => /lib64/libgcc_s.so.1 (0x00007f0392347000)
        libc.so.6 => /lib64/libc.so.6 (0x00007f0391f7a000)
        /lib64/ld-linux-x86-64.so.2 (0x00007f0393f83000)

```

四、MySQL 实例创建：

多实例的部署：只需要替换 db_mysql_8030 的实例名称即可。

实例名规范：db_数据库名_端口_mysql、db_数据库名_端口_mariadb

```
1、创建/授权 mysql 的文件目录：
mkdir -p /data/database/db_mysql8030_mysql/{data,etc,logs,tmp}
chown mysql:mysql  /data/database/db_mysql8030_mysql/ -R
ll -d /data/database/db_mysql8030_mysql/{data,etc,logs,tmp}


2、创建 my.cnf 配置文件：
cat > /data/database/db_mysql8030_mysql/etc/my.cnf  << EOF
[client]
port = 3307
socket = /data/database/db_mysql8030_mysql/tmp/db_mysql8030.sock
default_character_set = utf8mb4

[mysql]
loose-skip-binary-as-hex
prompt="(\D)[\u@mysql][\d]>"
no-auto-rehash
default_character_set = utf8mb4

[mysqld]
user = mysql
port = 3307
server_id = 63307
report-host = 172.20.0.6
report-port = 3307
skip_name_resolve = ON
default_authentication_plugin = mysql_native_password
#authentication_policy = mysql_native_password
```

```
# dir path
basedir = /usr/local/mysql8030
datadir = /data/database/db_mysql8030_mysql/data
socket = /data/database/db_mysql8030_mysql/tmp/db_mysql8030.sock
pid-file = /data/database/db_mysql8030_mysql/tmp/db_mysql8030.pid
innodb_data_home_dir = /data/database/db_mysql8030_mysql/data
innodb_log_group_home_dir = /data/database/db_mysql8030_mysql/logs
secure_file_priv =


# sql mode
sql_mode = NO_ENGINE_SUBSTITUTION


# character set
character_set_server = utf8mb4
character_set_client_handshake = 0
init_connect = 'set names utf8mb4'
collation_server = utf8mb4_unicode_ci
skip_name_resolve = 1

# read only （ 从库设置为 1 ）
read_only = 0
super_read_only = 0

#若你的MySQL数据库主要运行在境外，请务必根据实际情况调整本参数
default_time_zone = "+8:00"



#performance setttings
max_allowed_packet = 64M
lock_wait_timeout = 30
open_files_limit = 65535
back_log = 1024
max_connections = 1024
max_connect_errors = 99999999
table_open_cache = 512
table_definition_cache = 512
thread_stack = 512K
sort_buffer_size = 4M
join_buffer_size = 4M
read_buffer_size = 8M
```

```
read_rnd_buffer_size = 4M
bulk_insert_buffer_size = 64M
thread_cache_size = 128
interactive_timeout = 500
wait_timeout = 500
tmp_table_size = 96M
max_heap_table_size = 96M

#log settings
log_timestamps = SYSTEM
log_error = /data/database/db_mysql8030_mysql/logs/db_mysql8030.err
log_error_verbosity = 3
slow_query_log = 1
log_slow_extra = 1
slow_query_log_file = /data/database/db_mysql8030_mysql/logs/db_mysql8030.sl
general_log_file = /data/database/db_mysql8030_mysql/logs/db_mysql8030.log
long_query_time = 1
log_queries_not_using_indexes = 0
log_throttle_queries_not_using_indexes = 60
min_examined_row_limit = 100000
log_slow_admin_statements = 1
log_slow_replica_statements = 1
# log_slow_slave_statements = 1
# log_slow_verbosity = FULL


# binlog & relaylog
log_bin = /data/database/db_mysql8030_mysql/data/db_mysql8030-bin
binlog_format = ROW
sync_binlog = 1
binlog_cache_size = 4M
max_binlog_cache_size = 4G
max_binlog_size = 512M
binlog_rows_query_log_events = 1
binlog_expire_logs_seconds = 604800
binlog_checksum = CRC32
relay_log = /data/database/db_mysql8030_mysql/data/db_mysql8030-relay-bin
relay_log_index = /data/database/db_mysql8030_mysql/data/db_mysql8030-relay-



# group commit
# binlog_group_commit_sync_no_delay_count = 10
# binlog_group_commit_sync_delay =  100


# gtid info
gtid_mode = ON
enforce_gtid_consistency = ON
log_replica_updates = ON

#myisam settings
key_buffer_size = 32M
myisam_sort_buffer_size = 64M



# M-S replication settings:
relay_log_recovery = 1
#master_info_repository=TABLE
#relay_log_info_repository=TABLE


# semi sync
#plugin_dir=/usr/local/mysql/lib/plugin
#plugin_load="rpl_semi_sync_master=semisync_master.so;rpl_semi_sync_slave=se
#rpl_semi_sync_master_enabled=1
#rpl_semi_sync_slave_enabled=1
#rpl_semi_sync_master_wait_no_slave=1
#rpl_semi_sync_master_timeout=1000
#rpl_semi_sync_master_wait_point=AFTER_SYNC




# parallel replication settings
binlog_transaction_dependency_tracking = WRITESET
# slave_preserve_commit_order = 1
replica_preserve_commit_order = 1
# slave_checkpoint_period = 2
replica_checkpoint_period = 2
## 8.0.27 之前默认是 DATABASE，即 5.6 版本库的并行复制机制。8.0.27之后默认改成 L
# slave_parallel_type = LOGICAL_CLOCK
# replica_parallel_type = LOGICAL_CLOCK
## 并行复制线程数可以设置为逻辑CPU数量的2倍
# slave_parallel_workers = 8
```

```
replica_parallel_workers = 8



# GreatSQL 版本启用InnoDB并行查询优化功能
#loose-force_parallel_execute = OFF
#设置每个SQL语句的并行查询最大并发度
#loose-parallel_default_dop = 8
#设置系统中总的并行查询线程数，可以和最大逻辑CPU数量一样
#loose-parallel_max_threads = 8
#并行执行时leader线程和worker线程使用的总内存大小上限，可以设置物理内存的5-10%左右
#loose-parallel_memory_limit = 512M

#mgr settings
loose-plugin_load_add = 'mysql_clone.so'
loose-plugin_load_add = 'group_replication.so'
# MGR 集群的 UUID:
#loose-group_replication_group_name = "aaaaaaaa-aaaa-aaaa-aaaa-aaaaaaaaaaa1"

# 新加入节点，clone全量数据时，选择的 donor 节点，尽量使用非主节点作为 donor 节点
#clone_valid_donor_list = "172.20.0.118:33071"

# MGR本地节点IP:PORT，请自行替换:
#loose-group_replication_local_address = "172.20.0.6:33071"

# MGR集群所有节点IP:PORT，请自行替换:
#loose-group_replication_group_seeds = '172.20.0.6:33071,172.20.0.118:33071,

# 数据库启动时是否需要自动拉起 MGR:
#loose-group_replication_start_on_boot = ON

# 数据库启动只有是否开启 MGR 引导节点:
#loose-group_replication_bootstrap_group = OFF

# MGR 节点出现异常时改成 read_only 模式:
#loose-group_replication_exit_state_action = READ_ONLY

# 禁用流控制模式:
#loose-group_replication_flow_control_mode = "DISABLED"

# 单主模式:
#loose-group_replication_single_primary_mode = ON

```

```
# MGR 通信的最大消息大小。大于此大小的消息将自动拆分为片段，分别发送并由收件人重新纟
#loose-group_replication_communication_max_message_size = 10M

# MGR 接收最大的事务大小：
#loose-group_replication_transaction_size_limit = 512MB


# 设置少数成员组到多数成员组的网络超时时间，如果不设置将会一直等待，其上事务会被阻塞
#loose-group_replication_unreachable_majority_timeout = 30


# 设置从怀疑的创建（在最初的5秒检测期之后发生）到成员被驱逐出组之间的间隔时间。最大
#loose-group_replication_member_expel_timeout = 5


# 使成员在此刻（怀疑等待期结束）自动尝试重新加入组。
#loose-group_replication_autorejoin_tries = 288


# GreatSQL 版本开启/关闭冲裁节点功能，
# 注意在有仲裁节点的情况下，将单主切换成多主模式时，需要把投票节点先关闭再机型切换，
#loose-group_replication_arbitrator = 0

# 可以对每个节点设置地理标签，主要用于解决多机房数据同步的问题。
# 作用：在同城多机房部署方案中，同一个机房的节点可以设置相同的数值，另一个机房里的节
# 作用：这样在事务提交时会要求每组 group_replication_zone_id 中至少有个节点确认事
#group_replication_zone_id = [0~8]

# GreatSQL 版本 启用关闭快速单主模式：
#loose-group_replication_single_primary_fast_mode = 1

# GreatSQL 版本 MGR 集群选主，优先判断各节点事务应用状态，自动选择拥有最新事务的节
#group_replication_primary_election_mode=GTID_FIRST

# GreatSQL 版本 记录那些因为网络延迟较大导致的MGR性能瓶颈（单位：毫秒）
#loose-group_replication_request_time_threshold = 20000

#innodb settings
innodb_buffer_pool_size = 512M
innodb_buffer_pool_instances = 8
innodb_data_file_path = ibdata1:1024M:autoextend
innodb_flush_log_at_trx_commit = 1
innodb_log_buffer_size = 32M
```

```
innodb_log_file_size = 512M
innodb_log_files_in_group = 4
innodb_doublewrite_files = 2
innodb_max_undo_log_size = 4G
# 根据您的服务器IOPS能力适当调整
# 一般配普通SSD盘的话，可以调整到 10000 - 20000
# 配置高端PCIe SSD卡的话，则可以调整的更高，比如 50000 - 80000
innodb_io_capacity = 200
innodb_io_capacity_max = 800
innodb_open_files = 65534
innodb_flush_method = O_DIRECT
innodb_lru_scan_depth = 1024
innodb_lock_wait_timeout = 10
innodb_rollback_on_timeout = 1
innodb_print_all_deadlocks = 1
innodb_online_alter_log_max_size = 4G
innodb_print_ddl_logs = 1
innodb_status_file = 1
innodb_status_output = 0
innodb_status_output_locks = 1
innodb_sort_buffer_size = 64M

#innodb monitor settings
innodb_monitor_enable = "module_innodb"
innodb_monitor_enable = "module_server"
innodb_monitor_enable = "module_dml"
innodb_monitor_enable = "module_ddl"
innodb_monitor_enable = "module_trx"
innodb_monitor_enable = "module_os"
innodb_monitor_enable = "module_purge"
innodb_monitor_enable = "module_log"
innodb_monitor_enable = "module_lock"
innodb_monitor_enable = "module_buffer"
innodb_monitor_enable = "module_index"
innodb_monitor_enable = "module_ibuf_system"
innodb_monitor_enable = "module_buffer_page"
innodb_monitor_enable = "module_adaptive_hash"

#pfs settings
performance_schema = 1
#performance_schema_instrument = '%memory%=on'
performance_schema_instrument = '%lock%=on'
EOF
```

```
3、初始化mysql：
ll /data/database/db_mysql8030_mysql/etc/my.cnf
chown mysql:mysql  /data/database/db_mysql8030_mysql -R
ll /data/database/db_mysql8030_mysql/etc/my.cnf

/usr/local/mysql8030/bin/mysqld --defaults-file=/data/database/db_mysql8030_

# 没有消息就是成功。
# 注意1、 --initialize-insecure:  初始化的 mysql 时，不带初始密码。自动化脚本安
# 注意2、 --initialize：初始化的 mysql 时，带有初始密码。需要到 error.log 里找到
  使用 shell 命令获取：cat /data/database/db_mysql8030_mysql/logs/db_mysql803



4、启动 mysql 进程：
/usr/local/mysql8030/bin/mysqld_safe --defaults-file=/data/database/db_mysql



5、登录 mysql 控制台初始化用户：
mysql -uroot -p -S /data/database/db_mysql8030_mysql/tmp/db_mysql8030.sock



6、初始化用户：
ALTER USER user() IDENTIFIED BY "xWEKquBk7MsyGpMz";
CREATE USER root@'127.0.0.1' IDENTIFIED with mysql_native_password BY 'xWEKq
CREATE USER root@'172.%.%.%' IDENTIFIED with mysql_native_password BY 'xWEKq
CREATE USER root@'10.%.%.%'  IDENTIFIED WITH mysql_native_password BY 'xWEKq
CREATE USER root@'192.%.%.%' IDENTIFIED with mysql_native_password BY 'xWEKq
CREATE USER root@'localhost' IDENTIFIED with mysql_native_password BY 'xWEKq
CREATE USER tools_user@'192.%.%.%' IDENTIFIED with mysql_native_password BY
CREATE USER tools_user@'172.%.%.%' IDENTIFIED with mysql_native_password BY
CREATE USER tools_user@'10.%.%.%' IDENTIFIED with mysql_native_password BY '
CREATE USER tools_user@'127.0.0.1' IDENTIFIED with mysql_native_password BY
CREATE USER tools_user@'localhost' IDENTIFIED with mysql_native_password BY
CREATE USER repl@'192.%.%.%' IDENTIFIED with mysql_native_password BY '12345
CREATE USER repl@'172.%.%.%' IDENTIFIED with mysql_native_password BY '12345
CREATE USER repl@'10.%.%.%' IDENTIFIED with mysql_native_password BY '123456
CREATE USER repl@'127.0.0.1' IDENTIFIED with mysql_native_password BY '12345
CREATE USER repl@'localhost' IDENTIFIED with mysql_native_password BY '12345
CREATE USER admin_op@'192.%.%.%' IDENTIFIED with mysql_native_password BY '1
CREATE USER admin_op@'172.%.%.%' IDENTIFIED with mysql_native_password BY '1
CREATE USER admin_op@'10.%.%.%' IDENTIFIED with mysql_native_password BY '12
CREATE USER admin_op@'127.0.0.1' IDENTIFIED with mysql_native_password BY '1
CREATE USER admin_op@'localhost' IDENTIFIED with mysql_native_password BY '1
flush privileges;
flush logs;
reset master;


8、查询用户
select user,host,authentication_string,plugin,grant_priv from mysql.user;
show grants for 'admin_op'@'127.0.0.1';
```

五、其他常用命令:

```

一、关闭:
mysqladmin -S /data/database/db_mysql_8030_mysql/tmp/mysql.sock -p shutdown
或者
mysql> shutdown;



二、创建myin/mygo文件
1、安装gzexe工具:
yum install gzexe
# gzexe -d myin: 解密db_list文件
# gzexe myin : 加密db_list文件

2、编辑 myin 登陆脚本
vim ~/bin/myin
#!/bin/bash
p=$1
shift
mysql -h"127.0.0.1" -P"$p" --default-character-set=utf8 --show-warnings -uroc



三、密码生成器:
```

```
1、md5sum anaconda-ks.cfg |head -c 20
2、yum install makepasswd -y
   makepasswd  --char 20
```