# mysql部署

#!/bin/bash  
#date:2021-11-23  
#desc: for mysql install  
#set -x  
script\_dir=$(cd $(dirname $0) && pwd)  
cd $script\_dir  
read -p "请输入安装绝对路径（如 /home/data）：" istdir  
  
test ! -d ${istdir} && mkdir ${istdir}  
  
blue() {  
 echo -e "\033[34m $1 \033[0m" && sleep 1  
}  
  
red() {  
 echo -e "\033[31m $1 \033[0m" && sleep 1  
}  
  
istMysql () {  
 #安装必要软件包  
 yum -y install numactl libaio bzip2  
 #创建相关目录  
 blue "创建相关目录"  
 mkdir -pv ${istdir}/mysql/{run,data,binlogs,log}  
  
   
 blue "判断是否存在安装包"  
 if [ -f mysql-5.7.32-linux-glibc2.12-x86\_64.tar.gz ];then  
 echo -e "\033[32m mysql-5.7.32-linux-glibc2.12-x86\_64.tar.gz已存在 \033[0m"  
 else  
 echo -e "\033[32m mysql安装包不存在! 开始下载...... \033[0m" && wget https://cdn.mysql.com/archives/mysql-5.7/mysql-5.7.32-linux-glibc2.12-x86\_64.tar.gz  
 fi  
   
 blue "解压并mv到指定目录"  
 tar -xzvf mysql-5.7.32-linux-glibc2.12-x86\_64.tar.gz && mv mysql-5.7.32-linux-glibc2.12-x86\_64/\* ${istdir}/mysql  
   
 #添加环境变量  
 blue "添加环境变量"  
 echo "export PATH=$PATH:${istdir}/mysql/bin" >> /etc/profile  
 source /etc/profile  
   
 #添加用户组, 创建mysql用户添加到mysql组  
 blue "添加用户组, 创建mysql用户添加到mysql组"  
 groupadd mysql  
 useradd -g mysql mysql  
   
 #设置mysql文件夹权限  
 blue "设置mysql文件夹权限"  
 chown -R mysql.mysql ${istdir}/mysql/  
 #chown -R mysql.mysql /data/mysql/  
   
 #mysql的服务脚本放到系统服务中  
 blue "mysql的服务脚本放到系统服务中"  
 cp -a ${istdir}/mysql/support-files/mysql.server /etc/init.d/mysqld  
   
 #配置文件修改  
 blue "my.cnf配置文件修改"  
 rm -f /etc/my.cnf  
cat > /etc/my.cnf << EOF  
[client]  
#character\_set\_client = utf8  
port = 3306  
socket = /sqldir/mysql/run/mysql.sock  
  
[mysqld]   
port = 3306   
socket = /sqldir/mysql/run/mysql.sock  
basedir = /sqldir/mysql   
datadir = /sqldir/mysql/data   
pid-file = /sqldir/mysql/run/mysql.pid  
innodb\_file\_per\_table = ON   
log-error=/sqldir/mysql/log/mysql\_error.log   
lower\_case\_table\_names=1  
event\_scheduler = 1  
autocommit = 1  
character\_set\_server = utf8  
skip\_name\_resolve = 1  
max\_connections = 20000  
max\_connect\_errors = 100  
transaction\_isolation = READ-COMMITTED  
explicit\_defaults\_for\_timestamp = 1  
join\_buffer\_size = 8M #128GB  
tmp\_table\_size = 64M #128GB  
max\_allowed\_packet = 128M #128GB  
interactive\_timeout = 7200 #s  
wait\_timeout = 7200 #s  
read\_buffer\_size = 4M  
read\_rnd\_buffer\_size = 8M  
sort\_buffer\_size = 4M  
  
#slow\_query  
slow\_query\_log=ON  
slow\_query\_log\_file=/sqldir/mysql/log/mysql\_slow\_query.log  
long\_query\_time=2  
  
#log  
log\_queries\_not\_using\_indexes = 1  
log\_slow\_admin\_statements = 1  
log\_slow\_slave\_statements = 1  
log\_throttle\_queries\_not\_using\_indexes = 10  
min\_examined\_row\_limit = 100  
log\_timestamps=system  
  
########replication settings########  
master\_info\_repository = TABLE  
sync\_binlog = 1  
relay\_log\_recovery = 1b   
  
#innodb  
innodb\_flush\_log\_at\_trx\_commit = 2  
innodb\_buffer\_pool\_size = 72G #20-65% memory  
innodb\_buffer\_pool\_instances = 8  
innodb\_lru\_scan\_depth = 2000 #ssd下配置2000以上  
innodb\_lock\_wait\_timeout = 60  
innodb\_io\_capacity\_max = 8000 #ssd 8000  
innodb\_io\_capacity = 4000   
innodb\_flush\_method = O\_DIRECT   
innodb\_file\_format = Barracuda  
innodb\_file\_format\_max = Barracuda  
innodb\_flush\_neighbors = 0 #ssd  
innodb\_log\_file\_size = 140M  
innodb\_log\_buffer\_size = 16M  
innodb\_print\_all\_deadlocks = 1  
innodb\_strict\_mode = 1  
#innodb\_log\_group\_home\_dir = /sqldir/mysql/redolog/  
#innodb\_undo\_directory = /sqldir/mysql/undolog/  
#innodb\_undo\_log\_truncate=1  
#innodb\_max\_undo\_log\_size=2G  
  
  
  
server-id=1022  
sql\_mode = "STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION"  
  
  
#binlog  
log-bin=/sqldir/mysql/binlogs/mysql-bin.log  
log-slave-updates = 1   
binlog-format=row  
sync-master-info = 1  
sync\_binlog = 1 #ssd  
expire\_logs\_days = 10  
max\_binlog\_size = 100M   
log\_bin\_trust\_function\_creators=1  
binlog\_gtid\_simple\_recovery=1  
  
#GTID   
gtid-mode=on  
enforce-gtid-consistency=on  
master-info-repository=TABLE  
relay-log-info-repository=TABLE  
slave-parallel-workers=0  
binlog-checksum=CRC32  
master-verify-checksum=1  
slave-sql-verify-checksum=1  
binlog-rows-query-log\_events=1  
auto-increment-increment = 2  
auto-increment-offset = 1  
skip\_slave\_start=1  
log\_slave\_updates=1  
report-host=1.2.3.4  
report-port=33061  
EOF  
 totalmem=$(awk '/MemTotal/{total=$2}END{print (total)/1024/1024}' /proc/meminfo)  
 ibps=$(awk -v x=${totalmem} -v y=0.65 'BEGIN{printf "%.0f\n",x\*y}')  
 sed -i 's#innodb\_buffer\_pool\_size = 72G#innodb\_buffer\_pool\_size = '"$ibps"'G#g' /etc/my.cnf  
   
 ip=`ip a|grep inet|grep global|awk '{print $2}'|awk -F/ '{print $1}'`  
 sed -i 's#report-host=1.2.3.4#report-host='"$ip"'#g' /etc/my.cnf  
 sed -i 's#\/sqldir#'"$istdir"'#g' /etc/my.cnf  
   
   
 #查看mysql文件权限  
 blue "查看mysql文件权限"  
 ls -l ${istdir}/mysql/  
   
 #初始化数据库  
 blue "初始化数据库"  
 cd ${istdir}/mysql/bin/ && ./mysqld --initialize --user=mysql --basedir=${istdir}/mysql/--datadir=${istdir}/mysql/data/  
   
 #设置启动  
 blue "设置启动"  
 sleep 3  
 \cp -fv ${istdir}/mysql/support-files/mysql.server /etc/init.d/mysql && chmod +x /etc/init.d/mysql  
 chkconfig --add mysql && chkconfig --list mysql  
   
 #touch /data/mysql/mysqld.log && chown mysql.mysql /data/mysql/mysqld.log  
   
 #启动  
 blue "启动mysql"  
 service mysqld start  
 ps -ef|grep --color=auto mysql  
   
 #临时密码  
 blue "临时密码"  
 DBPWD=$(grep 'temporary password' ${istdir}/mysql/log/mysql\_error.log|awk '{print $NF}')  
 echo "$DBPWD" > /tmp/pwtemp.txt && echo -e "\033[32m mysql临时密码为:$DBPWD 并且 密码保存在/tmp/pwtemp.txt。\033[0m"  
   
 #mysql路径指定  
 blue "mysql路径指定"  
 ln -sfv ${istdir}/mysql/bin/mysql /usr/bin/mysql  
}  
  
mdmycnf () {  
cat >> /etc/my.cnf << EOF  
  
[mysqld\_safe]  
malloc-lib=/usr/lib64/libjemalloc.so.1  
EOF  
  
blue "重启mysql并且验证jemalloc是否生效"  
service mysqld restart  
lsof -n |grep jemalloc|grep mysql  
}  
  
#部署jemalloc  
istJem () {  
 blue "部署jemalloc"  
 if test -d /usr/local/include/jemalloc;then  
 blue "jemalloc已部署"  
 red "检查是否设置软链接"  
 if test -f /usr/lib64/libjemalloc.so.1;then  
 blue "/usr/local/lib/libjemalloc.so.2 已软链接到 /usr/lib64/libjemalloc.so.1"  
 red "jemalloc已完整部署" && mdmycnf && exit  
 else  
 red "未设置软链接,接下来设置！" && ln -sv /usr/local/lib/libjemalloc.so.2 /usr/lib64/libjemalloc.so.1  
 mdmycnf && exit  
 fi  
 fi  
   
   
 blue "安装 autogen autoconf"  
 yum -y install autogen autoconf  
   
 install () {  
 cd $script\_dir && tar -xf jemalloc-5.2.1.tar.bz2 && cd jemalloc-5.2.1  
 ./autogen.sh  
 make -j4  
 make install  
 ln -s /usr/local/lib/libjemalloc.so.2 /usr/lib64/libjemalloc.so.1  
 echo '/usr/local/lib' > /etc/ld.so.conf.d/local.conf  
 ldconfig  
 }  
   
 blue "下载and解压and部署"  
 if test -f jemalloc-5.2.1.tar.bz2;then   
 install   
 else   
 while true; do  
 #wget --no-check-certificate https://github.com/jemalloc/jemalloc/releases/download/5.2.1/jemalloc-5.2.1.tar.bz2  
 wget https://github.com/jemalloc/jemalloc/releases/download/5.2.1/jemalloc-5.2.1.tar.bz2 -O $script\_dir/jemalloc-5.2.1.tar.bz2  
 if [ "$?" != 0 ];then   
 red "未下载成功，将继续尝试下载，请耐心等待！"   
 else   
 blue "下载成功,开始部署！"  
 install && mdmycnf  
 break  
 fi  
 done  
 fi  
}  
  
main (){  
istMysql  
istJem  
}  
  
main "$@"