[mysqld]

#GENERAL

user=mysql

basedir=/opt/software/mysql80

datadir=/opt/data/mysql3306/data

port=3306

socket=/opt/data/mysql3306/mysql.socket

mysqlx\_port=33060

mysqlx\_socket=/opt/data/mysql3306/mysqlx.socket

log-error=/opt/logs/mysql3306/mysql80err.log

pid-file=/opt/data/mysql3306/mysql80.pid

default\_time\_zone="+08:00"

log\_timestamps=system

default\_storage\_engine=InnoDB

character\_set\_server=utf8mb4

transaction\_isolation=REPEATABLE-READ

activate\_all\_roles\_on\_login=1

tmpdir=/opt/data/mysql3306/tmp #用于控制 MySQL 服务器临时文件的存储路径。

secure\_file\_priv=/opt/data/mysql3306/tmp #用于控制 MySQL 服务器导入导出数据的操作权限，只基于文件

report\_host=’ip’

report\_port=

#LOGS

general\_log=0

general\_log\_file=/opt/logs/mysql3306/genaral/mysql3306.log

slow\_query\_log=1

slow\_query\_log\_file=/opt/logs/mysql3306/slow/slow.log

long\_query\_time=1

log\_queries\_not\_using\_indexes=1

#SESSION

#建议设置为内存的10%-25%，read\_rnd\_buffer\_size比read\_buffer\_size小

read\_buffer\_size=2M #控制用于顺序扫描表的读入缓冲区的大小1MB，该缓冲区用于存储从磁盘读取的数据

read\_rnd\_buffer\_size=4M #控制用于随机读取表的读入缓冲区的大小256KB，该缓冲区用于存储从磁盘读取的数据

sort\_buffer\_size=4M #控制用于排序操作的缓冲区大小256KB。该缓冲区用于存储需要排序的数据，以便 MySQL 可以更快地执行排序操作。

join\_buffer\_size=4M #控制用于连接表的连接缓冲区的大小512KB。该缓冲区用于存储连接操作所需的数据，以便 MySQL 可以更快地执行连接操作。

#CONNECTION

skip\_name\_resolve=1

back\_log=100 #控制 MySQL 服务器在停止接受新连接之前可以排队的连接数。50-65535建议为内存的10%-25%

max\_connections=100 #控制 MySQL 服务器可以同时接受的最大连接数。 151-65535建议为内存的10%-25%

max\_connect\_errors=10 #控制一个 IP 地址在连接到 MySQL 服务器之前允许的最大错误次数。10-2147483647

interactive\_timeout=1800 #控制服务器关闭交互式连接之前等待活动的秒数。

wait\_timeout=1800 #控制服务器等待客户端连接关闭的时间，单位为秒

thread\_cache\_size=128 #控制用于缓存空闲线程的线程缓存大小。该缓存用于存储不再使用的线程，以便 MySQL 可以更快地重用它们。10-65535建议为内存的10%-25%

max\_allowed\_packet=100M #控制服务器端和客户端在一次数据包传输过程中允许的最大数据包大小。如果数据包大小超过了设置的最大值，则会导致数据插入或更新失败。1M-1G建议为内存的10%-25%

#INNODB

innodb\_file\_per\_table=1

innodb\_data\_file\_path=ibdata1:1G:autoextend

innodb\_buffer\_pool\_size=512M

innodb\_buffer\_pool\_instances=4

innodb\_buffer\_pool\_load\_at\_startup=1 #控制服务器启动时是否将 InnoDB 缓冲池加载到内存中。

innodb\_buffer\_pool\_dump\_at\_shutdown=1 #控制服务器关闭时是否将 InnoDB 缓冲池的内容转储到磁盘。

#innodb\_log\_file\_size=512M

#innodb\_log\_files\_in\_group=2

innodb\_redo\_log\_capacity=1G

#innodb\_undo\_tablespaces=2

innodb\_max\_undo\_log\_size=1024M

innodb\_undo\_log\_truncate=1

innodb\_io\_capacity=200

innodb\_io\_capacity\_max=500

innodb\_read\_io\_threads=4

innodb\_write\_io\_threads=4

innodb\_page\_cleaners=8

innodb\_purge\_threads=4

innodb\_flush\_method=O\_DIRECT

innodb\_flush\_neighbors=1

innodb\_flush\_log\_at\_trx\_commit=1

innodb\_autoinc\_lock\_mode=2 #用于控制为 AUTO\_INCREMENT 列生成值时如何进行锁定的行为

innodb\_checksum\_algorithm=crc32 #用于指定用于计算 InnoDB 页校验和的算法。校验和用于验证数据页的完整性，并在数据库崩溃或损坏时帮助恢复数据。

innodb\_strict\_mode=1 #用于控制 InnoDB 引擎在执行某些操作时的严格程度。该参数可以帮助您提高数据完整性和一致性，但也可能会降低性能。值越高对性能影响越大，生产建议1

innodb\_print\_all\_deadlocks=1 #用于控制是否将所有死锁信息打印到错误日志中。

innodb\_numa\_interleave=1 #用于控制 InnoDB 引擎在 NUMA（非一致内存访问）架构上分配内存的方式。0为将内存分配在本地，1为将内存分配所有节点，可提高性能但加大内存使用量

innodb\_open\_files=65535 #控制 InnoDB 引擎可以同时打开的 .ibd 文件数量，包括数据文件、索引文件、重做日志文件等等

innodb\_adaptive\_hash\_index=OFF #用于控制 InnoDB 引擎是否自动创建自适应哈希索引。如果数据库负载较高，并且存在大量主键查询和唯一索引查询，可以打开

#REPLICATION

server\_id=1

log\_bin=/opt/logs/mysql3306/binlog/mysql-bin

relay\_log=/opt/logs/mysql3306/relaylog/relay-bin

sync\_binlog=1

binlog\_format=ROW

master\_info\_repository=TABLE

relay\_log\_info\_repository=TABLE

relay\_log\_recovery=ON #用于在主从复制环境中保证数据一致性的参数。当从库发生故障后，relay\_log\_recovery 可以帮助从库自动恢复数据。

工作原理

relay\_log\_recovery 通过在从库上存储主服务器的二进制日志来实现。当从库启动时，它会从主服务器下载二进制日志，并将其应用到自己的数据库中。如果从库发生故障，relay\_log\_recovery 会自动恢复从库上次成功应用的二进制日志的位置。然后，从库会从主服务器下载丢失的二进制日志，并将其应用到自己的数据库中。

#log\_slave\_updates=ON

log\_replica\_updates=ON

binlog\_expire\_logs\_seconds=604800

slave\_rows\_search\_algorithms='INDEX\_SCAN,HASH\_SCAN' #用于控制从库在执行数据变更操作时如何查找数据行

slave\_net\_timeout=60

skip\_slave\_start=ON

binlog\_error\_action=ABORT\_SERVER

super\_read\_only=ON

#GTID

gtid\_mode=ON

enforce\_gtid\_consistency=ON

binlog\_gtid\_simple\_recovery=TRUE #用于控制从库在使用 GTID（全局事务标识符）进行复制时如何执行简单恢复。设置为 TRUE 时，从库在执行简单恢复时只会扫描最新的二进制日志文件，以找到所有已执行的事务。

#SEMI\_REPLICATION

plugin\_load='validate\_password.so;rpl\_semi\_sync\_master=semisync\_master.so;rpl\_semi\_sync\_slave=semisync\_slave.so'

rpl\_semi\_sync\_master\_enabled=ON

rpl\_semi\_sync\_slave\_enabled=ON

rpl\_semi\_sync\_master\_timeout=10000 #用于控制主库在等待从库确认事务时，超时等待的时间

#MULTI\_THREAD\_REPLICATION

slave\_parallel\_type=LOGICAL\_CLOCK

slave\_parallel\_workers=4

slave\_preserve\_commit\_order=ON #用于控制从库在执行数据变更操作时是否保留主库提交顺序

binlog\_transaction\_dependency\_tracking= WRITESET\_SESSION #用于控制主库在生成二进制日志（binlog）时如何记录事务之间的依赖关系

binlog\_transaction\_dependency\_history\_size=25000 #用于控制主库在使用 WRITESET 或 WRITESET\_SESSION 模式记录事务依赖关系时，内存中保存的依赖关系历史记录的最大数量。值越大，从库并行复制性能越高，主库内存消耗越大

transaction\_write\_set\_extraction=XXHASH64 #是 MySQL Group Replication 中的一个参数，用于控制事务写集提取的算法，事务写集是指事务修改的所有行集。XXHASH64 是一种快速、非加密的哈希函数，具有良好的性能和冲突检测能力。它是 64 位版本的 XXH 哈希函数，适用于 64 位系统。

#OTHERS

open\_files\_limit=65535 #用于控制 MySQL 服务器可以同时打开的文件数量，包括表文件、索引文件、临时文件等等。

max\_heap\_table\_size=32M #用于控制 MEMORY 引擎表 的最大允许大小。

tmp\_table\_size=32M #用于控制 临时表 的最大允许大小,过低导致临时表无法存储足够数据，影响性能，过高浪费内存

table\_open\_cache=65535 #用于控制 MySQL 服务器可以同时打开的表文件数量，可以使用ulimit -n查看系统打开最大文件数限制，设置比它高即可

table\_definition\_cache=65535 #用于控制 MySQL 服务器可以缓存的表定义数量。根据 show status like '%table\_definition\_cache%',查看命中率进行调整

table\_open\_cache\_instances=64 #用于控制表缓存实例的数量。根据SHOW GLOBAL STATUS LIKE 'Table\_open\_cache\_instances%'查看使用情况

[mysqldump]

quick