

現有產品的 config 配置

```
[client]
port                = 3306
socket              = /var/run/mysql/mysql.sock
#socket=/var/lib/mysql/mysql.sock

# [mysqld_safe]
# socket            = /var/run/mysql/mysql.sock
# nice               = 0

[mysqld]
user                = mysql
pid-file            = /var/run/mysql/mysql.pid
socket              = /var/run/mysql/mysql.sock
# socket=/var/lib/mysql/mysql.sock

port                = 3306
bind-address        = 0.0.0.0
basedir             = /usr

datadir=/var/lib/mysql
# datadir           = /usr/local/mysql
tmpdir              = /tmp
lc-messages-dir     = /usr/share/mysql
# skip-external-locking

#key_buffer          = 16M
max_allowed_packet  = 1024M
#thread_stack        = 192K
#max_connections     = 5000
#table_cache         = 64
#thread_concurrency  = 128
#query_cache_limit   = 1M
#query_cache_size    = 16M
query_cache_type     = 1
max_connections=1800

#key_buffer          = 16M
max_allowed_packet  = 1024M
#thread_stack        = 192K
#max_connections     = 5000
#table_cache         = 64
#thread_concurrency  = 128
#query_cache_limit   = 1M
#query_cache_size    = 16M
query_cache_type     = 1
max_connections=1800

log-error=/var/log/mysql.log
#log_error           = /var/log/mysql_error.log
#log_slow_queries    = /var/log/mysql/mysql-slow.log
#long_query_time     = 2
#
sql_mode = STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION
# STRICT_TRANS_TABLES: 如果一個值不能插入到一個事務表中，則中斷當前的操作；對非事務表不做限制。
# NO_ZERO_IN_DATE: 在嚴格模式下，不允許日期和月份為零。
# NO_ZERO_DATE: 不允許插入零日期，插入零日期會噴出錯誤而不是警告。
# ERROR_FOR_DIVISION_BY_ZERO: 在INSERT或UPDATE過程中，如果資料被零除(分母為零)，則噴出錯誤而非警告。
# NO_ENGINE_SUBSTITUTION: 如果需要的儲存引擎被禁用或未編譯，那就噴錯誤。不設定此值時，用預設的儲存引擎替代，並噴出一個異常。

# 5.7版本的 MySQL 預設的 sql_mode 是: ONLY_FULL_GROUP_BY, STRICT_TRANS_TABLES, NO_ZERO_IN_DATE, NO_ZERO_DATE, ERROR_FOR_DIVISION_BY_ZERO, NO
# 其中ONLY_FULL_GROUP_BY和STRICT_TRANS_TABLES是從5.7.5版本加進來的,NO_AUTO_CREATE_USER是從5.7.7版加進來的,ERROR_FOR_DIVISION_BY_ZERO, NO_ZERO_D

#log-queries-not-using-indexes

# 需要才打開 slow log
slow_query_log =1
slow_query_log_file = /usr/local/mysql/mysql-rp01-slow.log
long_query_time = 2
#
# BinLog
server-id         = 1
# 若為 Slave 機器則要修改 server-id
log_bin          = /var/lib/mysql/mysql-bin.log
```

```

# log-slave-updates = 0
expire_logs_days = 10
max_binlog_size = 100M
innodb_flush_log_at_trx_commit = 1
# 每次事務提交時都會將日誌緩存區中的日誌寫入os cache並呼叫fsync()刷新到磁盤上的redo log file。
# (資料安全性高，因為只要提交就會更新磁盤資料所以不會有丟失的狀況，不過因為每次提交都刷新磁盤所以效能相對最差)
sync_binlog = 1
# 只有 master 設定為 1, Slave 與備份機都是設定為 0
# 寫緩衝x次，刷一次磁碟(0:預設系統每隔一段時間重整快取資料到磁碟,1:每次事務提交就會寫入磁碟,N:每N個事務提交...)
read_only = 0

#####
## 2個參數sync_binlog/innodb_flush_log_at_trx_commit 都為預設1的狀況下
## 作用: 保證(Server層)binlog與(引擎層)redo log的數據一致性。
#####

# innodb tune (optional)
innodb_change_buffering = all
# 緩存所有的操作, all = default
innodb_adaptive_hash_index = ON
# The adaptive hash index (AHI) lets InnoDB perform more like an in-memory database on systems with appropriate combinations of workload.
# adaptive hash index (AHI) 使 InnoDB 平台看起來更像一個內存資料庫 (在系統負載時還過得去時,且可供分配的 buffer pool 緩存池的容量充裕情況下), 且不犧牲任何
# ON = default
innodb_max_dirty_pages_pct = 80
# 控制 buffer pool 中髒頁的百分比, 當髒頁數量 % 超過參數值時, InnoDB 會啟動刷新髒頁的操作。
# 此參數指控制 %, 不影響刷新髒頁的速度。
# 最小 = 0, 最大 = 99.99, default = 75。
innodb_buffer_pool_size = 17179869184
# 這配置了緩衝區的記憶體大小, 將直接影響讀寫的效率。
# 當然是越多越好! 數據讀寫在內存中非常快, 減少了對硬碟的讀寫, 當資料 commit 或滿足 checkpoint 條件後才一次性把內存資料刷新到硬碟中。
# 緩衝區記憶體還會有操作系統或資料庫排程工作會使用;
# 根據經驗, 推薦此參數為 Server 總可用的緩衝區記憶體的 75%。
innodb_buffer_pool_chunk_size = 17179869184
# default = 128M
# innodb_buffer_pool_chunk_size 的最大值估算式: MAX(innodb_buffer_pool_chunk_size) = innodb_buffer_pool_size / innodb_buffer_pool_instances
innodb_buffer_pool_instances = 1
# buffer_pool 實例數量, default = 1。
innodb_print_all_deadlocks = 1
# 將所有的 deadlock logs 記錄到 errorlog 中

#[mysqldump]
#quick
#quote-names
#max_allowed_packet = 16M

#!includedir /etc/mysql/conf.d/
#skip-grant-tables
#
# 密碼原則
#validate_password_length=6
#validate_password_policy=LOW

# audit log
#server_audit_events = 'CONNECT,QUERY, TABLE, QUERY_DDL, QUERY_DML_NO_SELECT, QUERY_DCL'
# CONNECT: 記錄成功與失敗連線, 還有中斷連線。此事件包含使用者資訊。
# QUERY: 以純文字記錄所有查詢, 包括因為語法或許可錯誤而失敗的查詢。
# TABLE: 記錄因為執行查詢而受影響的資料表。
# QUERY_DDL: 類似於 QUERY 事件, 但只傳回資料定義語言 (DDL) 查詢 (CREATE·ALTER 等)。
# QUERY_DML_NO_SELECT: 類似於 QUERY 事件, 但只傳回資料操作語言 (DML) 查詢 (INSERT·UPDATE 和 SELECT 等), 但不記錄SELECT查詢。
# QUERY_DCL: 類似於 QUERY 事件, 但只傳回資料控制語言 (DCL) 查詢 (GRANT·REVOKE 等)。
#server_audit_logging = on
# 啟動審計功能
#server_audit_file_path = '/usr/local/mysql/mysql-audit-log/'
#server_audit_file_rotate_size = 200M
# 限制文件的大小
#server_audit_file_rotations = 20
# 要保存的文件數
#server_audit_file_rotate_now = ON
# 強制文件輪轉, 切新檔的意思

slave_parallel_workers = 16
# 平行處理總數 | Elvis 建議可以 CPU/2 來決定

slave_pending_jobs_size_max = 134217728
# 每次 pending 工作不會超過 128M

slave_parallel_type = 'LOGICAL_CLOCK'
# 以 timestamp 順序來執行平行處理

```

```
binlog_transaction_dependency_tracking = 'COMMIT_ORDER'
# 開啟平行處理的 default

slave_preserve_commit_order = 1
# 依照 master 來的順序執行平行處理

log_slave_updates = 1
# slave_preserve_commit_order 如果開啟，則 log_slave_updates 必須為 1

slave_transaction_retries = 10
# 重試次數 | 10 = default

master_info_repository = 'TABLE'
# slave 紀錄 master 的狀態，以 table 方式記錄。則資料庫中會有資料表 mysql.slave_master_info

relay_log_info_repository = 'TABLE'
# slave 同步 master 的訊息紀錄，以 table 方式記錄。則資料庫中會有資料表 mysql.slave_relay_info

relay_log_recovery = 1
# 從上次斷線時間重新抄寫 relay-log
```