

- `server-id`: MySQL 主从服务器上不能一样, 这是同一组主从结构的唯一标识。
- `log-bin`: 开启二进制日志 (搭建主从复制必须开启)。
- `binlog_format`: 二进制日志的格式, 有 `statement` 模式 (基于 SQL 语句的复制)、`row` 模式 (基于行的复制)、还有 `mixed` 模式 (混合复制), 这里必须使用 `row` 模式。statement 基于 SQL 语句的复制就是记录 SQL 语句在 binlog 中, 缺点是在某些情况下会导致主从节点中的数据不一致 (比如 `sleep()`、`now()` 等)。row 基于行的复制是 MySQL master 将 SQL 语句分解为基于 row 更改的语句并记录在 binlog 中, 也就是只记录哪条数据被修改了、修改成什么样, 优点是不会出现某些特定情况下被正确复制的问题。mixed 是以上两种模式的混合。
- `read_only`: 设置从库只读模式, 可以限定普通用户进行数据修改的操作, 但不会限定具有 `super` 权限的用户的数据修改操作, 可以通过 `set global read_only=1` 设置从库只读状态。MySQL 5.7 增加了一个 `super_read_only` 参数, 一旦开启该参数, 连超级管理员都没有权限进行写入操作。在 MySQL slave 库中设定了 `read_only=1` 以后, 通过 `show slave status\G` 命令查看 slave 状态, 发现 slave 仍然会读取 master 上的日志, 并且在 slave 库中应用日志, 不会影响 slave 同步复制的功能。
- `relay_log_recovery=1`: 当 slave 从库宕机后, 若 relay log 损坏了, 导致一部分中继日志没有处理, 则自动放弃所有未执行的 relay log, 并重新从 master 上获取日志, 这样就保证了 relay log 的完整性。默认情况下, 该功能是关闭的, 将 `relay_log_recovery` 的值设置为 1 时, 可在 slave 从库上开启该功能。建议开启。
- `relay-log-info-repository=TABLE` 和 `master-info-repository=TABLE`: 在 MySQL 运行过程中宕机的话, 从库启动后必须能够恢复到已经执行事务的位置, 该信息传统上是存在文件中的, 有可能存在不一致或者损坏的风险。从 MySQL 5.7 开始, 可以用表来存储这些信息, 并把这些表设置为 InnoDB 引擎, 通过使用事务型存储引擎来恢复这个信息。
- `gtid_mode`: 是否开启 gtid 模式。若使用 gtid 模式, 则设置 `gtid_mode=on`。
- `enforce-gtid-consistency`: `enforce_gtid_consistency` 默认为 off, 可选 [off/on], 表示限定事务安全的 SQL 才允许被记录。例如, `create table...select` 语句以及 `create temporary table` 语句不被允许执行。( `create table...select` 会被拆分为两个事务, 比如 `create table` 和 `insert` 事务, 会导致相同的 GTID 分配给两个事务, 从库会忽略。)
- `log_slave_updates`: 通常情况下, 从服务器从主服务器接收到的更新不记入它的二进制日志。该选项的作用是将主服务器上获取数据变更的信息记录到 slave 的二进制日志文件中。对于级联复制  $A \rightarrow B \rightarrow C$ , 也就是说, A 为从服务器 B 的主服务器, B 为从服务器 C 的主服务器。为了能工作, B 必须既为主服务器又为从服务器。除了 A 和 B 启用二进制日志外, B 服务器必须启用 `log-slave-updates` 选项。另外, MySQL 5.6 的 GTID 复制模式也必须开启 `log_slave_updates` 参数, 否则启动就会报错, 因为需要在 bin-log 找到同步复制的信息。在 MySQL 5.7 里, 官方做了调整, 用一张 `gtid_executed` 系统表记录同步复制的信息, 可以不用开启 `log_slave_updates` 参数, 减少了从库的压力。