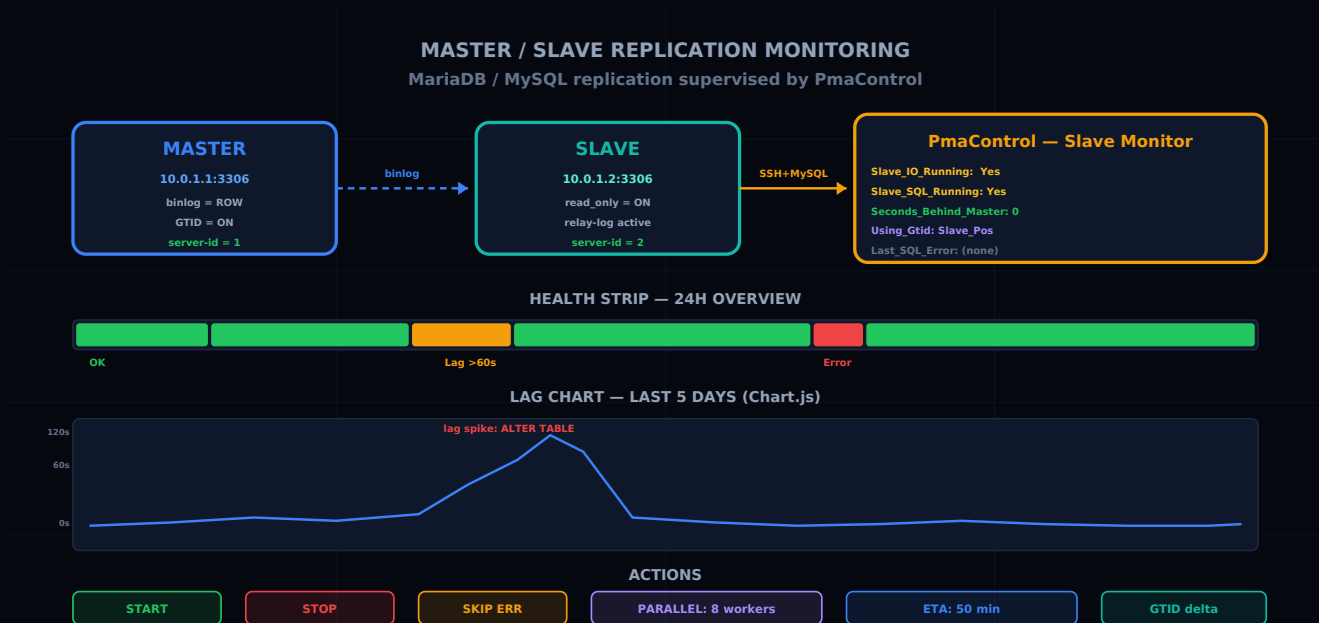


# Развёртывание Master/Slave и мониторинг репликации в PmaControl

Aurélien LEQUOY · April 13, 2026

MARIADB MYSQL REPLICATION MASTER-SLAVE MONITORING PMACONTROL



## Репликация в 2026 году: по-прежнему фундамент

Репликация master/slave остаётся фундаментальным строительным блоком любой production-инфраструктуры MariaDB / MySQL. Высокая доступность, распределённое чтение, горячие бэкапы — всё основано на ней. И тем не менее, большинство серьёзных инцидентов с базами данных связаны со сломанной репликацией или необнаруженным лагом.

Эта статья охватывает обе стороны репликации: как правильно её настроить, а затем как мониторить в PmaControl, чтобы никогда не быть застигнутым врасплох.

## Настройка репликации

### Предварительные требования на стороне мастера

Мастер должен иметь включённый binlog и уникальный server-id :

```
[mysqld]
server-id = 1
log-bin = /var/log/mysql/mysql-bin
binlog-format = ROW
gtid_strict_mode = 0N          # MariaDB
# enforce_gtid_consistency = 0N # MySQL
# gtid_mode = 0N              # MySQL
```

Создание пользователя репликации:

```
CREATE USER 'repl'@'10.0.1.%' IDENTIFIED BY 'secret_replication_password';
GRANT REPLICATION SLAVE ON *.* TO 'repl'@'10.0.1.%';
```

## Предварительные требования на стороне слейва

Слейву нужен собственный `server-id` и relay log:

```
[mysqld]
server-id = 2
relay-log = /var/log/mysql/relay-bin
read-only = 0N
log-slave-updates = 0N
```

`log-slave-updates` необходим, если вы планируете выстраивать слейвы в цепочку (слейв от слейва) или использовать Galera.

## Запуск репликации

### С GTID (рекомендуется)

На MariaDB:

```
CHANGE MASTER TO
  MASTER_HOST = '10.0.1.1',
  MASTER_USER = 'repl',
  MASTER_PASSWORD = 'secret_replication_password',
  MASTER_USE_GTID = slave_pos;

START SLAVE;
```

На MySQL 8.0+:

```
CHANGE REPLICATION SOURCE TO
  SOURCE_HOST = '10.0.1.1',
  SOURCE_USER = 'repl',
  SOURCE_PASSWORD = 'secret_replication_password',
  SOURCE_AUTO_POSITION = 1;

START REPLICA;
```

## Без GTID (классический режим)

Если GTID не активирован, необходимо зафиксировать позицию binlog мастера:

```
-- На мастере
SHOW MASTER STATUS;
-- File: mysql-bin.000042, Position: 154

-- На слейве
CHANGE MASTER TO
  MASTER_HOST = '10.0.1.1',
  MASTER_USER = 'repl',
  MASTER_PASSWORD = 'secret_replication_password',
  MASTER_LOG_FILE = 'mysql-bin.000042',
  MASTER_LOG_POS = 154;

START SLAVE;
```

## Проверка работоспособности

```
SHOW SLAVE STATUS\G
```

Два ключевых индикатора:

```
Slave_IO_Running: Yes
Slave_SQL_Running: Yes
Seconds_Behind_Master: 0
```

Если `Slave_IO_Running` равен `No`, слейв не может подключиться к мастеру (сеть, учётные данные, firewall). Если `Slave_SQL_Running` равен `No`, слейв не может применить события

(SQL-ошибка, нарушение ограничения).

## Добавление серверов в PmaControl

### Через веб-интерфейс

1. Перейдите в **Серверы -> Добавить сервер**
2. Укажите IP, порт (3306), учётные данные SSH и MySQL
3. PmaControl автоматически определяет роль (мастер или слейв) после первого цикла сбора данных

### Через REST API

```
# Добавить мастер
curl -X POST -H "Authorization: Bearer $TOKEN" \
  -H "Content-Type: application/json" \
  -d '{"ip": "10.0.1.1", "port": 3306, "name": "db-prod-master", "ssh_key_id": 1}' \
  https://pmacontrol.example.com/api/v1/servers

# Добавить слейв
curl -X POST -H "Authorization: Bearer $TOKEN" \
  -H "Content-Type: application/json" \
  -d '{"ip": "10.0.1.2", "port": 3306, "name": "db-prod-slave-01", "ssh_key_id": 1}' \
  https://pmacontrol.example.com/api/v1/servers
```

PmaControl запускает цикл сбора данных в течение минуты. После этого слейв отображается с бейджем "Slave" на дашборде.

## Страница Slave в PmaControl

Мониторинг репликации доступен по маршруту:

```
{lang}/slave/show/{id}/{name}/
```

Например: `/fr/slave/show/42/db-prod-slave-01/`

Контроллер `Slave.php` предоставляет два основных действия:

- **show()** : основная страница с текущим состоянием и графиком лага

- `showGraphDay()` : AJAX-данные для загрузки дополнительного дня на графике

## Отслеживаемые переменные

PmaControl собирает все переменные из `SHOW SLAVE STATUS` и сохраняет их в таблицы временных рядов. Наиболее критические:

Переменная	Значение
<code>Slave_IO_Running</code>	Активен ли поток IO (подключение к мастеру)
<code>Slave_SQL_Running</code>	Активен ли поток SQL (применение событий)
<code>Seconds_Behind_Master</code>	Лаг в секундах
<code>Using_Gtid</code>	Используемый режим GTID (MariaDB)
<code>Auto_Position</code>	GTID auto-position (MySQL)
<code>Last_SQL_Error</code>	Последняя встреченная SQL-ошибка
<code>Relay_Log_Space</code>	Размер текущего relay log

Для MySQL 8.0+ PmaControl также обрабатывает переименованные эквиваленты:

Старая переменная	Новая переменная (MySQL 8.0+)
<code>Slave_IO_Running</code>	<code>Replica_IO_Running</code>
<code>Slave_SQL_Running</code>	<code>Replica_SQL_Running</code>
<code>Seconds_Behind_Master</code>	<code>Seconds_Behind_Source</code>

PmaControl автоматически нормализует оба имени внутри системы.

## График лага

График лага — сердце страницы slave. Он отображает **последние 5 дней** значений `Seconds_Behind_Master` в виде кривой Chart.js.

Характеристики:

- **Разрешение:** одна точка в минуту (1440 точек в день)

- **Прогрессивная загрузка:** текущий день загружается немедленно, предыдущие дни — через AJAX «Загрузить предыдущий день»
- **Автоматический масштаб:** ось Y адаптируется к максимальному наблюдаемому лагу
- **Зелёная зона:** < 10 секунд, нормальная работа
- **Жёлтая зона:** 10-60 секунд, умеренный лаг
- **Красная зона:** > 60 секунд, критический лаг

## Полоса здоровья (Health Strip)

Над графиком горизонтальная полоса отображает состояние каждой минуты за 24 часа цветовым кодом:

Цвет	Условие
Зелёный	IO Running + SQL Running + лаг < 60 сек
Жёлтый	IO Running + SQL Running + лаг > 60 сек
Красный	IO или SQL остановлены, или критическая ошибка

Это мгновенный визуальный индикатор: одного взгляда достаточно, чтобы понять, был ли последний день стабильным или хаотичным.

## Корректирующие действия из PmaControl

Страница slave не ограничивается наблюдением. PmaControl позволяет действовать непосредственно на репликацию.

### START / STOP SLAVE

Две кнопки для запуска или остановки репликации. STOP SLAVE полезен для:

- Проведения обслуживания на слейве (тяжёлый ALTER TABLE)
- Создания консистентного бэкапа (снимок с остановленной репликацией)
- Диагностики проблемы лага (остановка для изучения binlog)

### SKIP error

Когда репликация останавливается из-за SQL-ошибки (дублирующее ограничение, отсутствующая таблица), PmaControl предлагает **пропустить событие**:

```
SET GLOBAL sql_slave_skip_counter = 1;  
START SLAVE;
```

**Внимание:** это действие требует явного подтверждения. Пропуск события означает согласие с расхождением между мастером и слейвом. PmaControl логирует действие с указанием пользователя, даты и пропущенной ошибки для отслеживаемости.

## Параллельные воркеры (параллельная репликация)

PmaControl предлагает **слайдер** для настройки количества воркеров параллельной репликации:

- **Минимум:** 1 (классическая последовательная репликация)
- **Максимум:** 50 или CPU x 2 (меньшее из двух)

```
STOP SLAVE;  
SET GLOBAL slave_parallel_workers = 8;  
START SLAVE;
```

Увеличение числа воркеров позволяет быстрее ликвидировать лаг, поскольку несколько транзакций применяются параллельно. Но это работает хорошо только с параллельной репликацией по LOGICAL\_CLOCK (MariaDB) или WRITESSET (MySQL 8.0+).

## Алгоритм ETA догоняния

Когда слейв имеет лаг, немедленный вопрос: **сколько времени до полного догоняния?**

PmaControl вычисляет оценку (ETA) на основе:

1. Текущий лаг в секундах
2. Наблюдаемая скорость догоняния (изменение лага за последние 10 минут)
3. Линейная экстраполяция

```
Если лаг уменьшился с 3600 с до 3000 с за 10 минут:  
Скорость = 600 с нагнано / 10 мин = 60 с/мин  
ETA = 3000 с / 60 с/мин = 50 минут
```

ETA отображается в верхней части страницы slave с прогресс-баром. Если скорость догоняния нулевая или отрицательная (лаг растёт), PmaControl показывает «ETA: расходящийся» красным цветом — сигнал, что необходимо вмешаться.

## Поддержка GTID

PmaControl автоматически определяет режим GTID:

- **MariaDB:** чтение `Using_Gtid` из `SHOW SLAVE STATUS`. Возможные значения: `No`, `Slave_Pos`, `Current_Pos`
- **MySQL:** чтение `Auto_Position` из `SHOW SLAVE STATUS`. Возможные значения: `0`, `1`

Когда GTID активен, PmaControl отображает дополнительную информацию:

- GTID set мастера ( `Gtid_Slave_Pos` или `Executed_Gtid_Set` )
- GTID set слейва
- Дельта между ними (количество транзакций отставания)

Дельта GTID часто более показательна, чем `Seconds_Behind_Master`: она даёт точное количество транзакций для догоняния, независимо от длительности каждой транзакции.

## Лучшие практики

### 1. Всегда использовать GTID

Режим позиции binlog (файл + позиция) хрупок: слишком рано удалённый файл, failover — и репликация сломана. GTID идемпотентен и переживает failover.

### 2. Включать `read_only` на слейвах

```
SET GLOBAL read_only = ON;  
SET GLOBAL super_read_only = ON; -- MySQL 5.7.8+ / MariaDB 10.3.16+
```

Без `read_only` случайная запись на слейв вызывает тихое расхождение данных.

### 3. Мониторить лаг, а не только состояние

`Slave_IO_Running: Yes` и `Slave_SQL_Running: Yes` недостаточно. Слейв может «работать», но с 2 часами лага. PmaControl отслеживает оба показателя: состояние И лаг.

## 4. Настроить оповещения

В PmaControl настройте пороги оповещений:

- **Warning:** лаг > 60 секунд в течение 5 минут
- **Critical:** лаг > 300 секунд ИЛИ IO/SQL остановлены

Оповещения отправляются через Telegram с указанием имени сервера, текущего лага и прямой ссылки на страницу slave.

## 5. Планировать тяжёлые операции

`ALTER TABLE` на объёмных таблицах генерируют временный пик лага. Используйте `pt-online-schema-change` или `gh-ost` для DDL в production и предупредите PmaControl, переводя слейв в режим обслуживания для предотвращения ложных срабатываний.

## Заключение

---

Репликация MariaDB / MySQL проста в настройке, но сложна в долгосрочном поддержании. PmaControl закрывает разрыв между «репликация работает» и «репликация здорова», предоставляя:

- Представление лага в реальном времени с историей за 5 дней
- Встроенные корректирующие действия (start/stop, skip, параллельные воркеры)
- Оценку догоняния (ETA) для прогнозирования
- Автоматическое определение GTID
- Проактивные оповещения через Telegram

Цель не в том, чтобы заменить DBA — а в том, чтобы дать ему инструменты для реагирования за минуты вместо часов.