

Podłączenie Telegram do PmaControl: alerty w czasie rzeczywistym

Aurélien LEQUOY · April 13, 2026

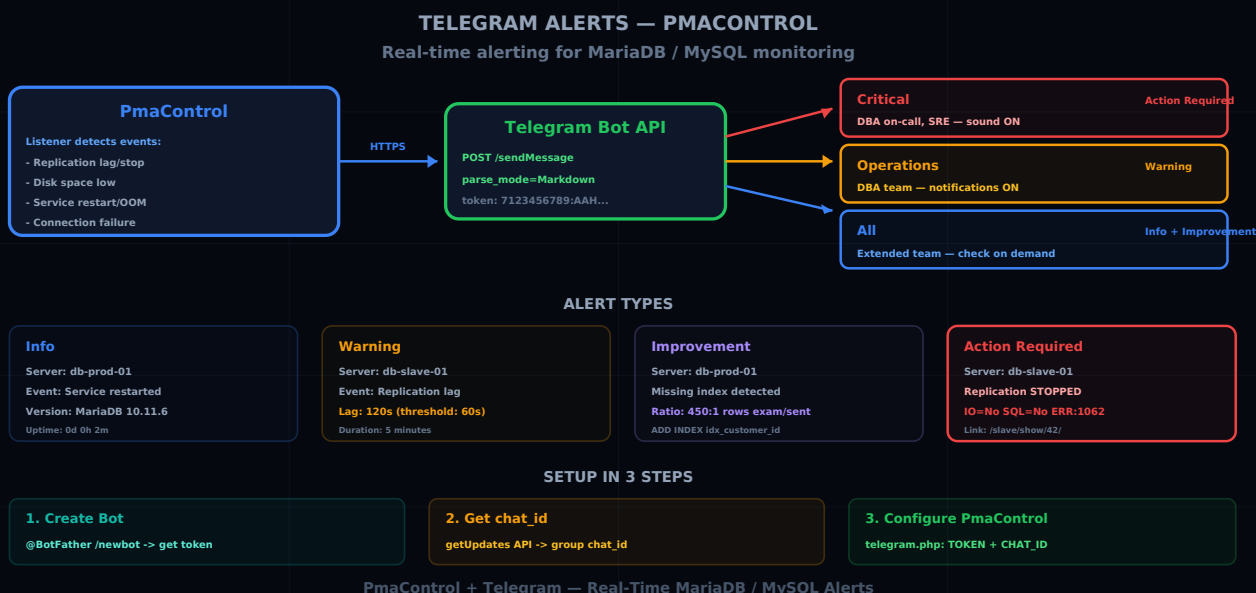
PMACONTROL

TELEGRAM

ALERTING

NOTIFICATIONS

AUTOMATION



Dlaczego Telegram, a nie email?

Alerty email mają fundamentalny problem: nikt nie czyta ich w czasie rzeczywistym. Email z alertem o 3 w nocy utonie w porannym potoku wiadomości. Alert Telegram wibruje w kieszeni dyżurnego DBA — opóźnienie między incydem a reakcją spada z godzin do sekund.

PmaControl używa Telegram jako głównego kanału alertów. To nie gadżet: to świadomy wybór techniczny. Boty Telegram są darmowe, API jest proste, wiadomości obsługują Markdown, a grupy pozwalają na routing alertów po ważności.

Tworzenie bota Telegram

Krok 1: Skontaktuj się z @BotFather

Otwórz Telegram i wyszukaj `@BotFather`. To oficjalny bot Telegram do tworzenia innych botów.


```
curl -s -X POST "https://api.telegram.org/bot7123456789:AAHxxxx/sendMessage" \  
-d chat_id=-1001234567890 \  
-d parse_mode=Markdown \  
-d text="*Test* : PmaControl alert system is working."
```

Jeśli wiadomość pojawi się w grupie, konfiguracja jest funkcjonalna.

Konfiguracja PmaControl

Plik konfiguracyjny

Konfiguracja Telegram w PmaControl znajduje się w:

```
/srv/www/pmacontrol/configuration/telegram.php
```

```
<?php  
// configuration/telegram.php  
  
define('TELEGRAM_TOKEN', '7123456789:AAHxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx');  
define('TELEGRAM_CHAT_ID', '-1001234567890');  
  
// Optional: separate channels by severity  
define('TELEGRAM_CHAT_ID_CRITICAL', '-1001234567891');  
define('TELEGRAM_CHAT_ID_WARNING', '-1001234567890');  
define('TELEGRAM_CHAT_ID_INFO', '-1001234567892');
```

Ważne: ten plik zawiera sekret (token). Zastosuj odpowiednie uprawnienia:

```
chmod 640 /srv/www/pmacontrol/configuration/telegram.php  
chown root:www-data /srv/www/pmacontrol/configuration/telegram.php
```

Typy alertów

PmaControl wysyła cztery typy alertów, każdy z poziomem ważności:

Info

Alerty informacyjne powiadamiają o normalnych, ale godnych uwagi zdarzeniach:

```
□ Info – PmaControl
Server: db-prod-01 (10.0.1.1:3306)
Event: Service restarted
Version: MariaDB 10.11.6
Uptime: 0 days 0 hours 2 minutes
Time: 2026-04-13 14:32:15
```

Przykłady zdarzeń Info:

- Restart usługi MariaDB / MySQL
- Wykrycie zmiany wersji
- Nowy serwer dodany do monitoringu
- Zakończony eksport schematu

Warning

Ostrzeżenia sygnalizują sytuacje wymagające obserwacji:

```
△□ Warning – PmaControl
Server: db-prod-slave-01 (10.0.1.2:3306)
Event: Replication lag exceeds threshold
Lag: 120 seconds (threshold: 60s)
Status: IO=Yes, SQL=Yes
Duration: 5 minutes
Time: 2026-04-13 14:32:15
```

Przykłady zdarzeń Warning:

- Opóźnienie replikacji > konfigurowalny próg
- Przestrzeń dyskowa < 20%
- Liczba połączeń > 80% maksimum
- Wzrost slow queries

Improvement

Sugestie ulepszeń generowane przez Marina+ lub wewnętrzne reguły:

```
□ Improvement – PmaControl
Server: db-prod-01 (10.0.1.1:3306)
```

```
Suggestion: Missing index detected
Query: SELECT * FROM orders WHERE customer_id = ?
Rows examined/sent ratio: 450:1
Recommended: ALTER TABLE orders ADD INDEX idx_customer_id (customer_id)
```

Action Required

Alerty krytyczne wymagające natychmiastowej interwencji:

```
🔴 Action Required – PmaControl
Server: db-prod-slave-01 (10.0.1.2:3306)
Event: Replication stopped
Slave_IO_Running: No
Slave_SQL_Running: No
Last_SQL_Error: Error 'Duplicate entry' on table 'users'
Time: 2026-04-13 14:32:15
Link: https://pmacontrol.example.com/fr/slave/show/42/
```

Przykłady zdarzeń Action Required:

- Replikacja zatrzymana (wątek IO lub SQL)
- Serwer nieosiągalny (nieudane połączenie SSH lub MySQL)
- OOM killer wykryty w logach
- Przechylenie dyskowa < 5% (krytyczna)
- Trwały błąd łączności (>3 próby)

RBAC: routing po ważności

W produkcji nie wszystkie alerty mają tych samych odbiorców. PmaControl obsługuje routing po ważności do różnych grup Telegram.

Zalecana architektura

```
Grupa "PmaControl - Critical" ← tylko Action Required
Członkowie: dyżurny DBA, lead tech, SRE

Grupa "PmaControl - Operations" ← Warning + Action Required
Członkowie: zespół DBA, SRE
```

Grupa "PmaControl - All" ← Info + Warning + Improvement + Action Required
Członkowie: rozszerzony zespół, główni developerzy

Konfiguracja

```
// configuration/telegram.php

// Grupa główna (wszystkie poziomy)
define('TELEGRAM_CHAT_ID', '-1001234567890');

// Grupy po ważności
define('TELEGRAM_CHAT_ID_CRITICAL', '-1001234567891'); // Action Required
define('TELEGRAM_CHAT_ID_WARNING', '-1001234567890'); // Warning
define('TELEGRAM_CHAT_ID_INFO', '-1001234567892'); // Info + Improvement

// Włącz routing po ważności
define('TELEGRAM_ROUTE_BY_SEVERITY', true);
```

Gdy `TELEGRAM_ROUTE_BY_SEVERITY` jest ustawione na `true`, każdy alert jest wysyłany wyłącznie do grupy odpowiadającej jego ważności. Grupa główna (`TELEGRAM_CHAT_ID`) otrzymuje kopię wszystkiego.

Zdarzenia alertowe

Restart usługi

PmaControl wykrywa restarty przez zmienną `Uptime`. Jeśli uptime spada z X godzin do kilku minut między dwoma zbiorami danych, to restart:

```
if ($current_uptime < $previous_uptime) {
    Telegram::sendInfo(
        "Service restarted",
        $server,
        "Previous uptime: " . formatDuration($previous_uptime) .
        "\nCurrent uptime: " . formatDuration($current_uptime)
    );
}
```

Opóźnienie replikacji

Opóźnienie jest monitorowane ciągle. Alert jest wyzwalany, gdy opóźnienie przekracza próg **przez konfigurowalny czas** (nie przy pojedynczym skoku):

```
// Alert tylko jeśli opóźnienie > próg od 5 minut
if ($lag > $threshold && $lag_duration > 300) {
    Telegram::sendWarning(
        "Replication lag exceeds threshold",
        $server,
        "Lag: {$lag}s (threshold: {$threshold}s)\n" .
        "Duration: " . formatDuration($lag_duration)
    );
}
```

Przestrzeń dyskowa

PmaControl zbiera dane o użyciu dysku przez SSH. Progi:

- **Warning:** < 20% wolnego
- **Critical:** < 5% wolnego

```
📧 Action Required – PmaControl
Server: db-prod-01 (10.0.1.1:3306)
Event: Disk space critical
Partition: /var/lib/mysql
Usage: 96% (used 458GB / 480GB)
Free: 22GB
```

OOM Killer

PmaControl analizuje syslog i dmesg w celu wykrycia zdarzeń OOM:

```
📧 Action Required – PmaControl
Server: db-prod-01 (10.0.1.1:3306)
Event: OOM Killer invoked
Process killed: mysqld (PID 1234)
Memory at kill: RSS 14.2GB, limit 16GB
Time: 2026-04-13 03:42:15
```

Łączność

Jeśli PmaControl nie może połączyć się z serwerem (timeout SSH lub odmowa połączenia MySQL):

```
☐ Action Required – PmaControl
Server: db-prod-03 (10.0.1.3:3306)
Event: Connection failed
Error: SSH timeout after 30s
Attempts: 3/3 failed
Last successful: 2026-04-13 14:25:00 (7 min ago)
```

Alert jest wysyłany dopiero po 3 nieudanych próbach, aby uniknąć fałszywych alarmów spowodowanych chwilowymi problemami sieciowymi.

Wysyłanie spersonalizowanych raportów

Agenci PmaControl mogą wysyłać sformatowane wiadomości w Markdown:

```
// Raport tygodniowy
$report = "*Weekly Report – PmaControl*\n\n";
$report .= "☐ *Servers monitored*: 142\n";
$report .= "☐ *Healthy*: 138\n";
$report .= "⚠ *Warnings*: 3\n";
$report .= "☐ *Critical*: 1\n\n";
$report .= "*Top issues this week:*\n";
$report .= "1. db-prod-slave-03: lag spike (max 340s) – resolved\n";
$report .= "2. db-staging-01: disk 87% – cleanup scheduled\n";
$report .= "3. db-prod-02: 12 slow queries > 10s – indexes added\n";

Telegram::send($report, TELEGRAM_CHAT_ID, 'Markdown');
```

Format Markdown Telegram obsługuje:

- **Pogrubienie:** `*tekst*`
- *Kursywa:* `_tekst_`
- Kod inline: ``tekst``
- Bloki kodu: ````tekst````
- Linki: `[tekst](url)`

Integracja z Releem agent

Releem to zewnętrzny agent optymalizacji MariaDB / MySQL. Gdy jest zintegrowany z PmaControl, jego rekomendacje są wysyłane przez Telegram:

```
📧 Improvement – Releem via PmaControl
Server: db-prod-01 (10.0.1.1:3306)
Recommendation: Increase innodb_buffer_pool_size
Current: 4GB
Recommended: 8GB
Reason: Buffer pool hit ratio 91.2% (target: >99%)
Impact: Estimated 15% improvement in read performance
```

Integracja odbywa się na poziomie Listenera: gdy Releem wysyła rekomendację przez API, PmaControl ją przetwarza i kieruje na Telegram w standardowym formacie.

Dobre praktyki

1. Nie zalewaj grup

Konfiguruj realistyczne progi. Grupa Telegram otrzymująca 50 alertów dziennie będzie ignorowana. Celuj w:

- **Critical:** 0-2 wiadomości na tydzień (idealnie 0)
- **Warning:** maksymalnie 5-10 wiadomości dziennie
- **Info:** w osobnej grupie, którą ludzie przeglądają na żądanie

2. Dołączaj linki akcji

Każdy alert powinien zawierać bezpośredni link do odpowiedniej strony PmaControl. Dyżurny DBA klika i trafia bezpośrednio na dashboard problematycznego serwera.

3. Dodawaj kontekst

"Replication lag: 120s" jest mniej przydatne niż "Replication lag: 120s (was 5s an hour ago, threshold: 60s, 3rd alert this week)". Kontekst pomaga w priorytetyzacji.

4. Testuj bota regularnie

Wysyłaj codzienną wiadomość testową ("PmaControl heartbeat: all systems operational"). Jeśli wiadomość przestanie przychodzić, wiesz, że bot jest zepsuty ZANIM nastąpi prawdziwy incydent.

```
# Codzienny cron heartbeat
0 8 * * * curl -s -X POST "https://api.telegram.org/bot$TOKEN/sendMessage" \
-d chat_id=$CHAT_ID \
-d text="PmaControl heartbeat: $(date) – all systems operational"
```

5. Chroń token

- Nigdy nie commituj go do Git
- Przechowuj w pliku konfiguracyjnym z restrykcyjnymi uprawnieniami
- Używaj menedżera sekretów w produkcji
- Unieważnij i wygeneruj token ponownie w razie kompromitacji (/revoke w @BotFather)

6. Konfiguruj powiadomienia mobilne

Telegram pozwala konfigurować powiadomienia per grupa:

- **Critical:** powiadomienia włączone, dźwięk włączony, zawsze widoczne
- **Operations:** powiadomienia włączone, dźwięk wyłączony
- **All:** powiadomienia wyłączone (przeglądanie na żądanie)

Rozwiązywanie problemów

Bot nie wysyła wiadomości

1. Sprawdź token: `curl "https://api.telegram.org/bot$TOKEN/getMe"`
2. Sprawdź chat_id: `curl "https://api.telegram.org/bot$TOKEN/getUpdates"`
3. Sprawdź, czy bot jest członkiem grupy
4. Sprawdź uprawnienia pliku `telegram.php`
5. Sprawdź logi PHP pod kątem błędów połączenia

Podwójne wiadomości

Jeśli każdy alert przychodzi podwójnie, sprawdź, czy `TELEGRAM_ROUTE_BY_SEVERITY` jest poprawnie skonfigurowany. Bez routingu alert trafia do domyślnej grupy. Z routingiem trafia do grupy specyficznej. Jeśli obie grupy to ta sama grupa, wiadomość przychodzi dwa razy.

Rate limiting

API Telegram ogranicza do ~30 wiadomości na sekundę na bota. Jeśli PmaControl nadzoruje 200 serwerów i wszystkie mają problem jednocześnie, wiadomości mogą być throttlowane.

Rozwiązanie: grupuj alerty w paczki:

```
☐ Action Required – PmaControl (batch)
```

```
Multiple servers affected:
```

- db-prod-01: Connection failed
- db-prod-02: Connection failed
- db-prod-03: Connection failed

```
Possible cause: Network outage in DC-1
```

Podsumowanie

Telegram to idealny kanał alertowy dla PmaControl: czas rzeczywisty, mobilny, formatowanie Markdown, grupy po ważności. Konfiguracja jest prosta (10 minut) i rezultat jest natychmiastowy: incydenty MariaDB / MySQL są wykrywane i zgłaszane w czasie rzeczywistym, z kontekstem niezbędnym do działania.

Klasyczna pułapka to nad-alertowanie. Grupa Telegram zalana powiadomieniami jest gorsza niż brak alertów — ludzie ją wyciszają. Kalibruj progi, oddzielaj ważności i regularnie testuj, czy bot działa.