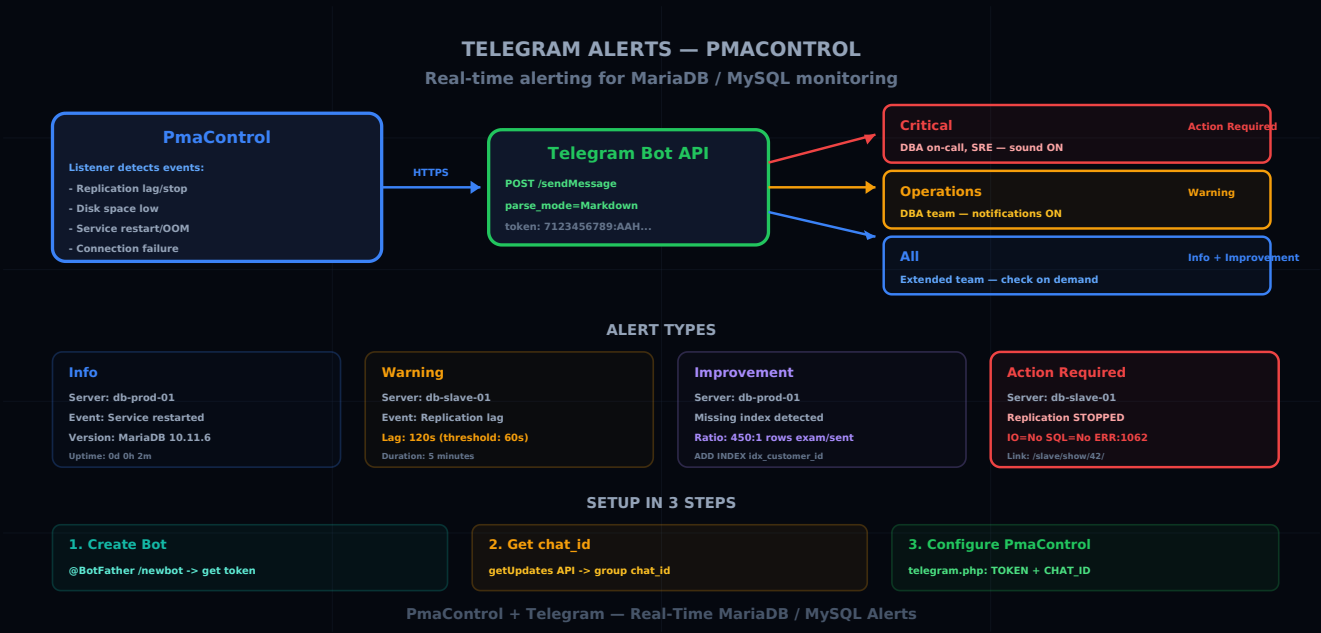
Brancher Telegram à PmaControl : alertes en temps réel

Aurélien LEQUOY · 13 avril 2026

PMACONTROL TELEGRAM ALERTING NOTIFICATIONS AUTOMATION



Pourquoi Telegram et pas email ?

Les alertes par email ont un problème fondamental : personne ne les lit en temps réel. Un email d'alerte à 3h du matin sera noyé dans le flux du lendemain. Une alerte Telegram vibre dans la poche du DBA d'astreinte — la latence entre l'incident et la réaction passe de heures à secondes.

PmaControl utilise Telegram comme canal d'alerte principal. Ce n'est pas un gadget : c'est un choix technique. Les bots Telegram sont gratuits, l'API est simple, les messages supportent le Markdown, et les groupes permettent de router les alertes par sévérité.

Créer un bot Telegram

Étape 1 : Contacter @BotFather

Ouvrez Telegram et cherchez `@BotFather`. C'est le bot officiel de Telegram pour créer d'autres bots.

```
Vous : /newbot
BotFather : Alright, a new bot. How are we going to call it?
Vous : PmaControl Alerts
BotFather : Good. Now let's choose a username for your bot.
Vous : pmacontrol_alerts_bot
BotFather : Done! [...] Use this token to access the HTTP API:
            7123456789:AAHxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

Conservez ce token — c'est la clé d'API de votre bot. Ne le commitez pas dans un dépôt Git, ne le partagez pas en clair.

Étape 2 : Obtenir le chat_id

Le bot a besoin de savoir à qui envoyer les messages. Pour un groupe :

1. Créez un groupe Telegram (ex: "PmaControl - Alertes Production")
2. Ajoutez votre bot au groupe
3. Envoyez un message dans le groupe
4. Appelez l'API `getUpdates` :

```
curl -s "https://api.telegram.org/bot7123456789:AAHxxxx/getUpdates" | python3 -m json.tool
```

Dans la réponse, cherchez le `chat_id` du groupe (il est négatif pour les groupes) :

```
{
    "result": [{
        "message": {
            "chat": {
                "id": -1001234567890,
                "title": "PmaControl - Alertes Production",
                "type": "supergroup"
            }
        }
    }]
}
```

Le `chat_id` est `-1001234567890`.

Étape 3 : Tester

```
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."
```

Si le message apparaît dans le groupe, la configuration est fonctionnelle.

Configurer PmaControl

Le fichier de configuration

La configuration Telegram de PmaControl se trouve dans :

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

```
<?php
// configuration/telegram.php

define('TELEGRAM_TOKEN', '7123456789:AAHxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx');
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');
```

Important : ce fichier contient un secret (le token). Appliquez les permissions appropriées :

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

Types d'alertes

PmaControl envoie quatre types d'alertes, chacun avec un niveau de sévérité :

Info

Les alertes informatives notifient des événements normaux mais notables :

```
□□ 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
```

Exemples d'événements Info :

- Redémarrage d'un service MariaDB / MySQL
- Changement de version détecté
- Nouveau serveur ajouté au monitoring
- Export de schéma terminé

Warning

Les avertissements signalent des situations à surveiller :

```
⚠□ 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
```

Exemples d'événements Warning :

- Lag de réplication > seuil configurable
- Espace disque < 20%
- Nombre de connexions > 80% du max
- Slow queries en augmentation

Improvement

Les suggestions d'amélioration sont générées par Marina+ ou les règles internes :

```
□□ 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

Les alertes critiques nécessitent une intervention immédiate :

```
□□ 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/
```

Exemples d'événements Action Required :

- Réplication stoppée (IO ou SQL thread)
- Serveur injoignable (connexion SSH ou MySQL échouée)
- OOM killer détecté dans les logs
- Espace disque < 5% (critique)
- Erreur de connectivité persistante (>3 tentatives)

RBAC : router par sévérité

En production, toutes les alertes n'ont pas la même audience. PmaControl supporte le routage par sévérité vers des groupes Telegram différents.

Architecture recommandée

```
Groupe "PmaControl - Critical"    ← Action Required uniquement
  Membres : DBA astreinte, lead tech, SRE

Groupe "PmaControl - Operations"  ← Warning + Action Required
  Membres : équipe DBA, SRE
```

```
Groupe "PmaControl - All"         ← Info + Warning + Improvement + Action Required
  Membres : équipe élargie, développeurs principaux
```

Configuration

```
// configuration/telegram.php

// Groupe principal (tous les niveaux)
define('TELEGRAM_CHAT_ID', '-1001234567890');

// Groupes par sévérité
define('TELEGRAM_CHAT_ID_CRITICAL', '-1001234567891');  // Action Required
define('TELEGRAM_CHAT_ID_WARNING',  '-1001234567890');  // Warning
define('TELEGRAM_CHAT_ID_INFO',     '-1001234567892');  // Info + Improvement

// Activer le routage par sévérité
define('TELEGRAM_ROUTE_BY_SEVERITY', true);
```

Quand `TELEGRAM_ROUTE_BY_SEVERITY` est `true`, chaque alerte est envoyée uniquement au groupe correspondant à sa sévérité. Le groupe principal (`TELEGRAM_CHAT_ID`) reçoit une copie de tout.

Événements d'alerte

Redémarrage de service

PmaControl détecte les redémarrages via la variable `Uptime`. Si l'uptime passe de X heures à quelques minutes entre deux collectes, c'est un redémarrage :

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

Lag de réplication

Le lag est surveillé en continu. L'alerte est déclenchée quand le lag dépasse le seuil **pendant une durée configurable** (pas un pic isolé) :

```
// Alerte seulement si le lag est > seuil depuis 5 minutes
if ($lag > $threshold && $lag_duration > 300) {
    Telegram::sendWarning(
        "Replication lag exceeds threshold",
        $server,
        "Lag: {$lag}s (threshold: {$threshold}s)\n" .
        "Duration: " . formatDuration($lag_duration)
    );
}
```

Espace disque

PmaControl collecte l'utilisation disque via SSH. Les seuils :

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

```
□□ 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 analyse le syslog et le dmesg pour détecter les événements 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
```

Connectivité

Si PmaControl ne peut pas se connecter à un serveur (SSH timeout ou MySQL connection refused) :

```
□□ 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)
```

L'alerte n'est envoyée qu'après 3 tentatives échouées pour éviter les faux positifs sur un glitch réseau.

Envoyer des rapports personnalisés

Les agents PmaControl peuvent envoyer des messages formatés en Markdown :

```
// Rapport hebdomadaire
$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');
```

Le format Markdown de Telegram supporte :

- **Gras** : `*texte*`
- *Italique* : `_texte_`
- `Code inline` : `` `texte` ``
- Blocs de code : ` ```texte``` `
- Liens : `[texte](url)`

Intégration avec Releem agent

Releem est un agent externe d'optimisation MariaDB / MySQL. Quand il est intégré à PmaControl, ses recommandations sont envoyées via 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
```

L'intégration se fait au niveau du Listener : quand Releem envoie une recommandation via l'API, PmaControl la traite et la route vers Telegram avec le format standard.

Bonnes pratiques

1. Ne pas noyer les groupes

Configurez des seuils réalistes. Un groupe Telegram qui reçoit 50 alertes par jour sera ignoré. Visez :

- **Critical** : 0-2 messages par semaine (idéalement 0)
- **Warning** : 5-10 messages par jour maximum
- **Info** : dans un groupe séparé que les gens consultent à la demande

2. Inclure les liens d'action

Chaque alerte devrait contenir un lien direct vers la page PmaControl correspondante. Le DBA d'astreinte clique et arrive directement sur le dashboard du serveur problématique.

3. Ajouter du contexte

"Replication lag: 120s" est moins utile que "Replication lag: 120s (was 5s an hour ago, threshold: 60s, 3rd alert this week)". Le contexte aide à prioriser.

4. Tester le bot régulièrement

Envoyez un message de test quotidien ("PmaControl heartbeat: all systems operational"). Si le message n'arrive plus, vous savez que le bot est cassé AVANT un vrai incident.

```
# Cron quotidien de 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. Protéger le token

- Ne jamais le commiter dans Git
- Le stocker dans un fichier de configuration avec permissions restrictives
- Utiliser un secret manager en production
- Révoquer et régénérer le token si compromis (`/revoke` sur @BotFather)

6. Configurer les notifications mobiles

Telegram permet de configurer les notifications par groupe :

- **Critical** : notifications activées, son activé, toujours visible
- **Operations** : notifications activées, son désactivé
- **All** : notifications désactivées (consultation à la demande)

Dépannage

Le bot n'envoie pas de messages

1. Vérifier le token : `curl "https://api.telegram.org/bot$TOKEN/getMe"`
2. Vérifier le chat_id : `curl "https://api.telegram.org/bot$TOKEN/getUpdates"`
3. Vérifier que le bot est membre du groupe
4. Vérifier les permissions du fichier `telegram.php`
5. Vérifier les logs PHP pour des erreurs de connexion

Messages en double

Si chaque alerte arrive en double, vérifiez que `TELEGRAM_ROUTE_BY_SEVERITY` est configuré correctement. Sans routage, l'alerte va au groupe par défaut. Avec routage, elle va au groupe

spécifique. Si les deux sont le même groupe, le message arrive deux fois.

Rate limiting

L'API Telegram limite à ~30 messages par seconde par bot. Si PmaControl supervise 200 serveurs et que tous ont un problème simultané, les messages peuvent être throttled. La solution : regrouper les alertes en batch :

```
□□ 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
```

Conclusion

Telegram est le canal d'alerte idéal pour PmaControl : temps réel, mobile, formatage Markdown, groupes par sévérité. La configuration est simple (10 minutes) et le résultat est immédiat : les incidents MariaDB / MySQL sont détectés et notifiés en temps réel, avec le contexte nécessaire pour agir.

Le piège classique est de sur-alerter. Un groupe Telegram noyé sous les notifications est pire que pas d'alertes du tout — les gens le mettent en silencieux. Calibrez les seuils, séparez les sévérités, et testez régulièrement que le bot fonctionne.