Alerting present and feature

Tibor Köröcz
Observability Tech Lead

PERCONA

© 2026 Percona

Agenda

01 Why Alerting Matters

02 Alert Mistakes

03 Recommendations

04 Feature of Alerting

PERCONA®

© 2026 Percona

Why Alerting Matters

Databases are the backbone of nearly every production system. When they degrade or fail, the business feels it immediately — in lost revenue, user churn, and eroded trust. A well-designed alerting system is your first line of defense.

⚡ **Reduce MTTR**

Fast, precise detection cuts mean time to recovery — often from hours to minutes

💰 **Protect Revenue**

Even brief downtime can cost thousands per minute in e-commerce and SaaS

🧠 **Empower Humans**

Alerts should deliver context that helps engineers make confident decisions fast

🔒 **Operational Confidence**

Teams that trust their alerts move faster and sleep better on-call

© 2026 Percona

The Biggest Alerting Failures

What Goes Wrong

Alert Fatigue

Hundreds of low-quality alerts desensitize on-call engineers, causing them to ignore or delay response

Threshold-Only Thinking

Static thresholds ignore workload patterns — a CPU spike at 2 AM is not the same as one at 2 PM

No Ownership or Escalation

Alerts that go to everyone often get actioned by no one — unclear ownership is a silent killer

The Real Cost

Missing Context at 3 AM

An alert that says "disk full" without a runbook, dashboard link, or impacted service is nearly useless under pressure

Monitoring Everything, Understanding Nothing

Instrumenting every metric without a signal-to-noise strategy leads to dashboards nobody trusts

Stale Alert Rules

Systems evolve, but alert rules rarely do

PERCONA®

© 2026 Percona

What Makes a Good Alert?

Every alert that fires should pass a simple test: **would an engineer know exactly what to do when they see this?** If the answer is no, the alert needs work. Use these five properties as your quality checklist.

Actionable

Every alert must have a clear, documented response — a runbook, a remediation step, or an escalation path

Timely

Early enough to react, but not so sensitive it fires on transient noise — sustained conditions matter

Context-Rich

Include service name, environment, dashboard URLs, and runbook links directly in the notification

Reliable

Low false-positive rate builds trust — engineers should never doubt whether an alert is real

Relevant

Tied to real business impact or service degradation — not just a metric crossing an arbitrary line

PERCONA®

© 2026 Percona

Too much context

{
  "details": {
    "firing": "
Value: A=-1, C=-1
Labels:
 - alertname = mysql_replication_lag
 - agent_id = 7903a5c7-5976-46d8-84d9-cabdec770353
 - agent_type = mysqld_exporter
 - cluster = ps-replication-dev-cluster
 - environment = ps-replication-dev
 - grafana_folder = Experimental
 - instance = 7903a5c7-5976-46d8-84d9-cabdec770353
 - job = mysqld_exporter_7903a5c7-5976-46d8-84d9-cabdec770353_mr
 - machine_id = 85a2cf29fa594a9e9a27aa88fce231ba
 - master_host = ps_pmm_replication_8_0_1
 - master_uuid = 8da41e19-0667-11f1-adeb-9247e0067a2c
 - node_id = a1cc8cae-dd58-41e6-86eb-12c866fa4e8c
 - node_name = 79e48ca85f38
 - node_type = generic
 - service_id = 01fb5b56-bb3e-4a33-9797-3b83306c6a20
 - service_name = ps_pmm_replication_8_0_2_84438
 - service_type = mysql
Annotations:
 - datasource_uid = PA58DA793C7250F1B
 - description = Whatever you want here
 - grafana_state_reason = NoData
 - ref_id = A
 - summary = Whatever you want here
Source:
https://localhost/graph/alerting/grafana/ffcx6mac9uoe8b/view?orgId=1
Silence: https://localhost/graph/alerting/silence/new?...
",
    "num_firing": "1",
    "num_resolved": "0"
  }
}

PERCONA®

© 2026 Percona

Add custom keys

Step 1: Navigate to Your PagerDuty Contact Point:

1. Go to **Alerting** → **Contact points**
1. Find your PagerDuty contact point and click **Edit**
1. Scroll down to the **Details** section

Step 2: Add Custom Key-Value Pairs

Click **+ Add** and create these key-value pairs:

firing {{ .CommonLabels.alertname }}
cluster {{ .CommonLabels.cluster }}
environment {{ .CommonLabels.environment }}
node_name {{ .CommonLabels.node_name }}
service_name {{ .CommonLabels.service_name }}
status {{ .Status }}

PERCONA

© 2026 Percona

Result

```
{
  "details": {
    "firing": "mysql_replication_lag",
    "cluster": "ps-replication-dev-cluster",
    "environment": "ps-replication-dev",
    "node_name": "79e48ca85f38",
    "service_name": "ps_pmm_replication_8_0_2_84438",
    "status": "firing",
    "num_firing": "1",
    "num_resolved": "0"
  }
}
```

PERCONA®

© 2026 Percona

Alerting Best Practices That Scale

Alert on Symptoms, Not Just Causes

Slow query response for users matters more than CPU being at 85%. Cause-only alerts miss the business impact entirely.

SLO/SLA-Driven Alerting

Anchor your alert thresholds to your Service Level Objectives — if your SLO is 99.9% availability, your alerts should defend exactly that contract.

Define Clear Severity Levels

P1 wakes people up. P3 creates a ticket. Ambiguous severity means inconsistent response and unnecessary escalations.

Continuously Review and Tune

Run a monthly alert review. Retire rules that never fire or always false-positive. Measure alert quality like you measure code quality.

© 2026 Percona

Reducing Alert Fatigue

Suppress Duplicates

Use deduplication keys so one infrastructure problem generates one page, not fifty

Multi-Condition Alerting

Combine signals: high latency AND elevated error rate together is more meaningful than either alone

Require Sustained Failure

A metric must breach threshold for 5–10 minutes before firing — eliminates transient spikes

Route by Ownership

DB alerts go to the DBA team, not a generic ops channel — right person, right alert, every time

Retire Noisy Alerts

Track alert frequency. Any rule firing more than 10 times a week without action is a candidate for retirement. If requires action fix the root cause.

© 2026 Percona

Percona PMM Alerting Overview

Percona Monitoring and Management (PMM) provides a purpose-built observability and alerting platform for MySQL, MongoDB, and PostgreSQL environments. Built on proven open-source foundations, it combines the power of Prometheus-style metrics with the richness of Grafana dashboards and a database-aware alert template library.

Prometheus-Compatible

Rules, recording rules, and PromQL expressions work natively

Alertmanager Routing

Full deduplication, grouping, silencing, and multi-channel delivery

Prebuilt DB Templates

Out-of-the-box alert rules for replication, connections, disk, and queries

Custom Rule Support

Author your own rules using PromQL and attach them to any exporter

PERCONA®

© 2026 Percona

Alerting on threads running is really the most important metric for your business ?

PERCONA®

© 2026 Percona

Alert on business impact

- We already monitor your database metrics
- Why not monitor your data?
- You can collect and aggregate:
  - **When was your last sale?**
  - **How much did you sell in last 5m/15m/1h?**

PERCONA®

© 2026 Percona

Custom Queries

```
---
mysql_orders_count:
  query: "SELECT COUNT(*) AS orders_count FROM orders"
  metrics:
    - orders_count:
        usage: "GAUGE"
        description: "Current number of orders."
```

More details:
https://www.percona.com/blog/pmms-custom-queries-in-action-adding-a-graph-for-innodb-mutex-waits/

PERCONA®

© 2026 Percona

Static thresholds are no longer enough.

Modern production databases run on dynamic, auto-scaling infrastructure with variable workloads, rolling deployments, and multi-cloud dependencies. A fixed number that was meaningful six months ago may be completely irrelevant today — or dangerously permissive tomorrow.

Dynamic Systems

Traffic patterns shift with features, campaigns, and seasonal load — no static threshold can keep pace

Anomaly Detection

ML models that learn baseline behavior catch subtle regressions that fixed thresholds completely miss

Metric Correlation

One anomaly means little. When five correlated signals spike together, that's a real incident worth waking someone up

Context Awareness

Alerts that understand deployment state, maintenance windows, and upstream dependencies eliminate entire classes of noise

PERCONA®

© 2026 Percona

How AI Will Change Alerting

The future of alerting is not hundreds of granular pages — it is a small set of high-level business impact alerts. When one fires, AI immediately launches a full investigation, correlating metrics, logs, traces, and events to turn a notification into a complete incident report.

Business Impact Signals

Instead of 100+ low-level alerts, AI surfaces a few meaningful alerts tied to customer experience, revenue, availability, and risk

Autonomous Investigation

The moment an alert fires, AI gathers evidence across metrics, logs, traces, and events to understand what actually changed

Root Cause + Context

Findings are attached directly to the alert with the likely cause, blast radius, and the evidence engineers need to trust the conclusion

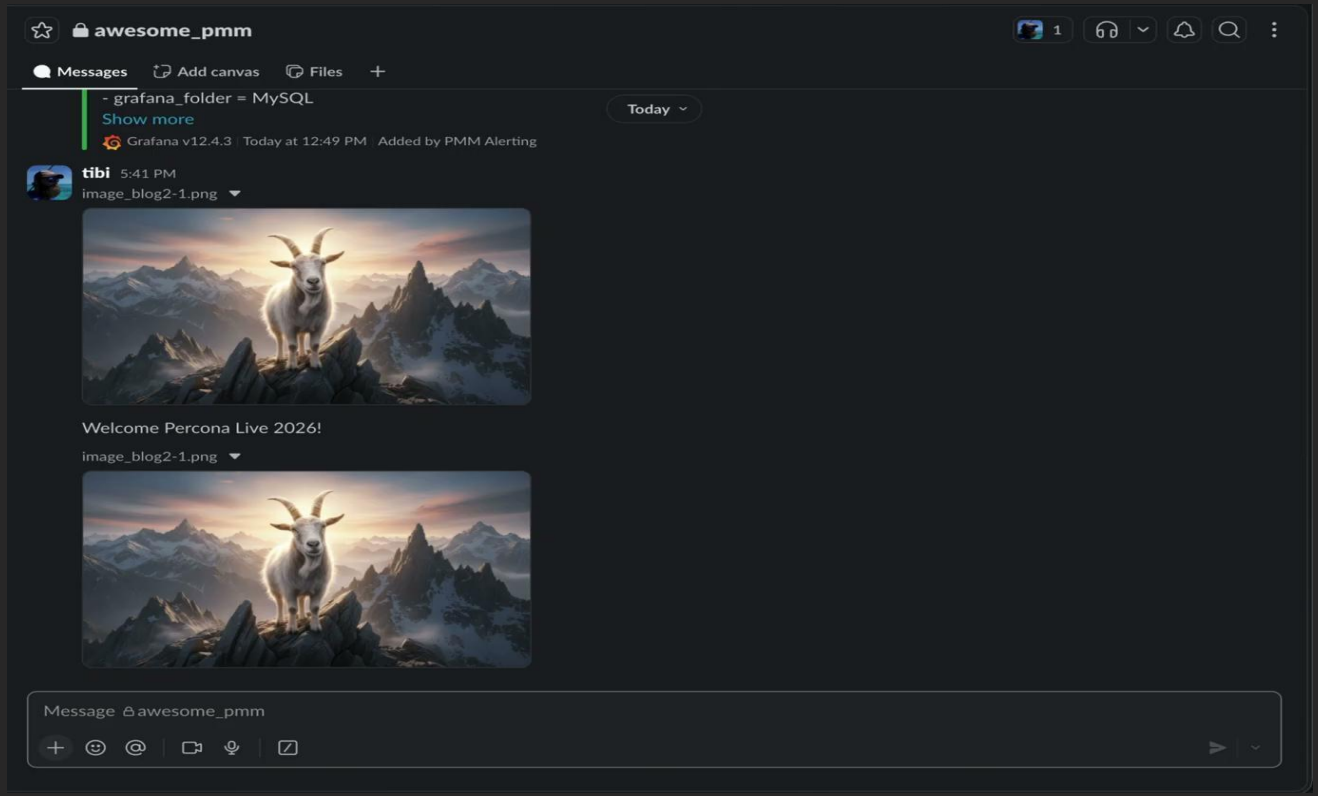
Recommended or Automated Remediation

The alert arrives with next steps — from suggested actions and runbooks to safe automation that can resolve known issues immediately

PERCONA®

© 2026 Percona

Sneak Peak



© 2026 Percona

Feature of Observability

Less Alerts - More actions

🔴 Reactive Today

"The database is down. Fix it NOW."

Engineers scramble, runbooks are pulled, users are already impacted. MTTD is measured in minutes of pain.

🟢 Predictive Tomorrow

"Anomaly detected in query latency — Auto-scaling triggered."

AI identifies the pattern, initiates remediation automatically, and notifies the SRE *before* the outage.

PERCONA®

© 2026 Percona

Conclusion: The Human Element

AI doesn't replace the SRE — it elevates them. Freed from alert firefighting, engineers can focus on architecture, resilience design, and meaningful improvements.

🎯 The Goal

Build systems that **heal themselves** — so your team can sleep through the night with confidence.

✅ Your Action This Week

Identify your **top 5 most frequent alerts**. For each one: Is it actionable? Can it be automated? Define a path forward.

PERCONA®

© 2026 Percona

PERCONA®

© 2026 Percona

© 2026 Percona