Yash Khokhar

Datadog

Contents

[Datadog Monitoring for ASP.NET-MySQL 2](#_Toc193108851)

[Install and Configure the Datadog Agent 2](#_Toc193108852)

[Prerequisites 2](#_Toc193108853)

[Method 1: GUI Installation 2](#_Toc193108854)

[Method 2: Command Line Installation 2](#_Toc193108855)

[Verify Installation 2](#_Toc193108856)

[Configure Agent 2](#_Toc193108857)

[Monitoring Your ASP.NET-MySQL Setup 3](#_Toc193108858)

[Monitor MySQL Metrics 3](#_Toc193108859)

[Set Alerts 3](#_Toc193108860)

[Visualize in Dashboards 3](#_Toc193108861)

[Enable Logs 3](#_Toc193108862)

[Detect Anomalies 4](#_Toc193108863)

[Correlate with System Metrics 4](#_Toc193108864)

[Trace Queries with APM 4](#_Toc193108865)

[Database Security Monitoring 5](#_Toc193108866)

[Capacity Planning & Forecasting 6](#_Toc193108867)

[Datadog vs. PMM for Your Setup 6](#_Toc193108868)

[Key Differences 6](#_Toc193108869)

[Recommendation 6](#_Toc193108870)

[Datadog Pricing 6](#_Toc193108871)

[Summary 7](#_Toc193108872)

# Datadog Monitoring for ASP.NET-MySQL

Datadog unifies monitoring and security for your ASP.NET-MySQL application on Windows, offering real-time insights into performance and issues at any scale. This guide covers setup, monitoring, tracing, and a comparison with Percona Monitoring and Management (PMM).

## Install and Configure the Datadog Agent

### Prerequisites

* **Datadog Account**: Sign up at [Datadog](https://www.datadoghq.com/) and get your API Key from **Integrations > APIs**.
* **Windows**: Server 2016+ or 10+ with Administrator privileges.

### Method 1: GUI Installation

1. **Download**: Get datadog-agent-7-latest.amd64.msi from [Datadog Agent Download Page](https://app.datadoghq.com/account/settings#agent).
2. **Install**: Run the .msi with Admin rights, accept terms, set path (C:\Program Files\Datadog\Datadog Agent\), input API Key, and finish.

### Method 2: Command Line Installation

1. **Open PowerShell as Admin**: Press Win + R, type powershell, press Ctrl + Shift + Enter.
2. **Run**: Replace <YOUR\_API\_KEY>:

* Start-Process -Wait msiexec -ArgumentList '/qn /i datadog-agent-7-latest.amd64.msi APIKEY="<YOUR\_API\_KEY>" TAGS="env:prod,app:aspnet"'
  + Options: SITE="datadoghq.com", /qn for silent, REBOOT=ReallySuppress to skip reboots.

### Verify Installation

1. **Check Status**:

* & "C:\Program Files\Datadog\Datadog Agent\bin\agent.exe" status
  + Expect confirmation of Agent running and metrics collection.

1. **Check UI**: In Datadog, go to **Infrastructure > Host Map**; confirm server visibility.

### Configure Agent

* Edit C:\ProgramData\Datadog\datadog.yaml: Set logs\_enabled: true, tags: ["env:prod", "app:aspnet"].
* Add MySQL: Create C:\ProgramData\Datadog\conf.d\mysql.d\conf.yaml:
* instances:  
   - host: 127.0.0.1  
   user: datadog  
   pass: password  
   port: 3306

## Monitoring Your ASP.NET-MySQL Setup

### Monitor MySQL Metrics

Datadog collects key MySQL metrics to spot issues like slow queries affecting API response times.

* **Query Performance**: mysql.performance.slow\_queries (lagging endpoints), mysql.performance.queries (traffic load).
* **Connections**: mysql.net.connections (active sessions), mysql.net.max\_connections (capacity limits).
* **Resources**: mysql.innodb.buffer\_pool\_utilization (memory), mysql.innodb.data\_reads (disk load). **Note**: Focus on queries and connections for ASP.NET; use dashboards/alerts to act.

### Set Alerts

Configure in **Monitors > New Monitor** to catch MySQL issues:

* mysql.performance.slow\_queries > 10 → Slack/Email for slow APIs.
* mysql.net.connections / mysql.net.max\_connections > 80% → Teams for session limits.
* mysql.performance.query\_run\_time > 5000ms → PagerDuty for long queries. **Destinations**: Email, Slack, Teams, PagerDuty, Webhooks.

### Visualize in Dashboards

Create real-time views in **Dashboards > New Dashboard**:

* **Query Trends**: **Timeseries Graph** for mysql.performance.query\_run\_time—spot slow endpoints.
* **Connections**: **Gauge** for mysql.net.connections vs. mysql.net.max\_connections—monitor session limits.
* **Top Queries**: **Top List** for mysql.performance.queries—find costly queries. **Tip**: Prioritize query and connection metrics; add resources if needed.

### Enable Logs

Capture logs to debug ASP.NET-MySQL issues:

1. Edit C:\ProgramData\Datadog\datadog.yaml:

* logs\_enabled: true

1. Update C:\ProgramData\Datadog\conf.d\mysql.d\conf.yaml:

* logs:  
   - type: file  
   path: "C:\mysql\logs\mysql-slow.log"  
   service: mysql  
   source: mysql  
   - type: file  
   path: "C:\mysql\logs\mysql-error.log"  
   service: mysql  
   source: mysql

1. Restart Agent:

* & "C:\Program Files\Datadog\Datadog Agent\bin\agent.exe" restart

**Tip**: Filter slow queries or errors in **Log Explorer**. Adjust paths to your MySQL log location.

### Detect Anomalies

Use machine learning in **Monitors > New Monitor** with anomalies() to flag odd behavior:

* **Query Slowdown**: 0.1s to 1s—fix slow APIs.
* **Traffic Spike**: 50 req/s to 300 req/s—scale or check attacks.
* **Connection Drop**: 20 to 5—restore access. **Why**: Keeps your app smooth by catching issues early.

### Correlate with System Metrics

Overlay MySQL with system stats in dashboards to find bottlenecks:

* **CPU & Memory**: system.cpu.user, system.mem.used—high usage slowing queries?
* **Disk I/O**: system.io.read, system.io.write—disk impacting load times?
* **Network**: system.net.bytes\_sent, system.net.bytes\_rcvd—delays in data transfer? **Tip**: Pair with mysql.performance.query\_run\_time for root causes.

### Trace Queries with APM

Datadog APM traces MySQL queries to ASP.NET code for optimization.

#### Benefits

* Spot slow queries (e.g., 5.6s delays) in endpoints.
* Identify high-impact queries (e.g., checkout).
* Fix inefficiencies with code context.

#### Steps

1. **Enable APM**: In C:\ProgramData\Datadog\datadog.yaml:

* apm\_config:  
   enabled: true
  + Restart: & "C:\Program Files\Datadog\Datadog Agent\bin\agent.exe" restart".

1. **Instrument ASP.NET**:
   * Add: dotnet add package Datadog.Trace.
   * Set vars: set DD\_TRACE\_ENABLED=true, set DD\_API\_KEY=<your\_api\_key>, set DD\_SERVICE=aspnet-app.
   * Run: dotnet run.
2. **View Traces**: In **APM > Traces**, filter by aspnet-app:

* SELECT \* FROM orders WHERE status = 'pending' → 5.6s

#### Example

[Route("api/orders")]  
public class OrdersController : ControllerBase  
{  
 private readonly MyDbContext \_context;  
 public OrdersController(MyDbContext context) => \_context = context;  
  
 [HttpGet("pending")]  
 public IActionResult GetPendingOrders()  
 {  
 var orders = \_context.Orders.Where(o => o.Status == "pending").ToList();  
 return Ok(orders);  
 }  
}

* **Output**: GET /api/orders/pending → 6s, with 5.6s MySQL—index status.

#### Notes

* Links to .NET code for debugging.
* Test overhead in staging for high traffic.

## Database Security Monitoring

Datadog tracks MySQL security risks for your ASP.NET-MySQL app via metrics and logs. Set alerts in **Monitors > New Monitor**:

* mysql.auth.failed—catches failed logins.
* root >5 queries/min—flags odd admin activity. **Tip**: Enable error logs (see Logs) to track security events.

## Capacity Planning & Forecasting

Forecast MySQL trends in dashboards with forecast():

* forecast(mysql.net.connections, 1w)—predicts session needs.
* forecast(mysql.innodb.data\_written, 30d)—plans storage.
* forecast(mysql.performance.queries, 1w)—anticipates traffic. **Tip**: Scale ASP.NET servers or MySQL resources before limits hit.

## Datadog vs. PMM for Your Setup

### Key Differences

| Feature | Datadog | PMM |
| --- | --- | --- |
| **Focus** | Full-stack (ASP.NET + MySQL) | MySQL only |
| **Tracing** | APM to .NET code | Query Analytics, no app |
| **Hosting** | Cloud SaaS | Self-hosted |
| **Setup** | Easy (Agent + NuGet) | Complex (Server + Client) |
| **Cost** | Paid | Free |

### Recommendation

* **Datadog**: Best for tracing MySQL queries (e.g., 5.6s in OrdersController) to ASP.NET, plus Windows stats and alerts. *Pros*: Easy setup, APM, AI anomalies. *Cons*: Paid, less query depth.
* **PMM**: Ideal for deep MySQL tuning (e.g., slow query logs) on a budget. *Pros*: Free, detailed metrics. *Cons*: No ASP.NET link, harder setup.
* **Hybrid**: Use PMM for tuning, Datadog for production.

## Datadog Pricing

Datadog offers several pricing tiers for Infrastructure Monitoring, which are billed annually per host:

* **Pro Plan**: $15 per host per month.
* **Enterprise Plan**: $23 per host per month.

For Application Performance Monitoring (APM), the pricing is:

* **APM Pro**: $35 per underlying APM host per month.
* **APM Enterprise**: $40 per underlying APM host per month, which includes Data Streams Monitoring and Continuous Profiler with four profiled containers per host per month.

For the most accurate and up-to-date information, please visit [Datadog Pricing](https://www.datadoghq.com/pricing/).

## Summary

Datadog equips your ASP.NET-MySQL app with end-to-end monitoring on Windows—install the Agent, track metrics, trace queries, and secure your stack. For deep MySQL focus, consider PMM alongside Datadog’s broader capabilities.