Monitoring ASP.NET Core Web API using Prometheus & Grafana

By
Narasimha Rao T
Microsoft.Net FSD Trainer
Professional Development Trainer
tnrao.trainer@gmail.com


K

Real World	Monitoring Equivalent
Theme Park	ASP.NET Core API
Sensors	Metrics (/metrics)
Security Guards	Prometheus
Control Room Screens	Grafana
Emergency Sirens	Alertmanager
Accidents	Errors (500)
Visitors	Requests
Ride Waiting Time	Response Time (Latency)

1. What is Prometheus?

Prometheus is an open-source **monitoring and alerting system** designed for modern applications.

Key points:

- Prometheus **collects metrics** (numbers describing system behavior).
- It **scrapes** (pulls) metrics from applications through an exposed HTTP endpoint (usually `/metrics`).
- Metrics follow a standard format called **Prometheus exposition format**.
- Prometheus stores metrics in a time-series database.
- Provides a powerful query language: **PromQL**.

2. Why Use Prometheus with ASP.NET Core?

- To monitor **API performance**: request rate, latency, error count.
- To monitor **system health**: CPU, memory usage (via exporters).
- Helps detect **production issues** before users notice.
- Works perfectly with **Docker**, **Kubernetes**, **microservices**, and cloud deployments.
- It is **free**, **lightweight**, and widely used in DevOps.

3. What is Grafana?

Grafana is a visualization tool used to create **dashboards** and **graphs** from various data sources (like Prometheus).

Key Points:

- Connects to Prometheus as a **data source**.
- Visualizes application performance: charts, heatmaps, alerts.
- Useful for **monitoring in real time**.
- Dashboards can be shared with teammates.

4. Why Use Grafana with Prometheus?

- Prometheus collects raw metrics → Grafana visualizes them beautifully.
- Easy to create dashboards for API performance.
- Helps track long-term system behavior.
- Alerts can be configured visually.

Prometheus vs Grafana: Monitoring vs Visualization Explained

In the lower section, the diagram visually representes the relationship between Promethaus and Grafana

Monitoring

Prometheus

- Monitoring system
- Metrins collection
- Alerting

Visualization

Grafana

- Visualization tool
- Dashboards
- Analytics

How to Use Prometheus & Grafana with ASP.NET Core Web API

Step 1: Install the Prometheus Middleware Package

Add this NuGet package:

```
dotnet add package prometheus-net.AspNetCore
```

Step 2: Expose Metrics Endpoint in ASP.NET Core

In **Program.cs**:

```
var builder = WebApplication.CreateBuilder(args);

builder.Services.AddControllers();

var app = builder.Build();

app.UseRouting();

// Enable Prometheus metrics
app.UseHttpMetrics(); // measures HTTP request metrics

app.MapControllers();

// Expose /metrics endpoint
app.MapMetrics();

app.Run();
```

Step 3: Install and Configure Prometheus

1. Download Prometheus:

https://prometheus.io/download/

2. Edit prometheus.yml:

```
global:
  scrape_interval: 5s

scrape_configs:
  - job_name: "aspnet_api"
    metrics_path: "/metrics"
    static_configs:
      - targets: ["host.docker.internal:5000"]
```

If not using Docker, use:

```
targets: ["localhost:5000"]
```

Step 4: Install Grafana

Download from:

https://grafana.com/grafana/download

Start Grafana and open:

```
http://localhost:3000
```

Default login:

- **username**: admin
- **password**: admin

Step 5: Add Prometheus as a Data Source in Grafana

1. Navigate to **Configuration → Data Sources**
2. Add **Prometheus**
3. Set URL:

   ```
   http://localhost:9090
   ```

4. Click **Save & Test**

Step 6: Create Dashboards

You can create panels that visualize:

- API requests per second
- Request duration
- Failed requests
- Memory usage
- CPU load

Or import ready-made dashboards:

- Grafana dashboard ID for .NET: **12464** (popular)

Example Metrics Prometheus Will Collect

Prometheus middleware automatically exposes:

- `http_requests_received_total`
- `http_request_duration_seconds`
- `process_cpu_seconds_total`
- `process_working_set_bytes` (memory)
- `dotnet_gc_*` (garbage collector performance)

Summary (Beginner Friendly)

Prometheus

- *What*: A monitoring & metrics collection tool.
- *Why*: Tracks API performance and system behavior.
- *How*: Add middleware, expose `/metrics`, configure Prometheus to scrape.

Grafana

- *What*: A dashboard & visualization tool.
- *Why*: Turns metrics into readable charts.
- *How*: Connect Grafana to Prometheus and build dashboards.

Together they give you **full observability** for your ASP.NET Core app.

Q & A

Narasimha Rao T
tnrao.trainer@gmail.com