DevOps Fundamentals - Observability - Logging Systems

[**🔍 1. Purpose** 1](#_Toc204939581)

[**📘 2. Theory** 1](#_Toc204939582)

[**🧰 3. Prerequisites** 2](#_Toc204939583)

[**🔧 4. Common Monitoring & Logging Tools** 2](#_Toc204939584)

[**🧪 5. Example: Monitoring with Azure Application Insights** 2](#_Toc204939585)

[**📸 6. Snapshot: Observability Workflow** 3](#_Toc204939586)

[**✅ 7. Summary** 3](#_Toc204939587)

**🔍 1. Purpose**

To understand how **Monitoring** and **Logging** play a critical role in maintaining **system reliability**, **performance**, and **security** in any DevOps lifecycle.

**📘 2. Theory**

**🔹 What is Monitoring?**

Monitoring involves **real-time tracking** of system metrics like CPU usage, memory, errors, response time, and uptime to **identify and resolve issues proactively**.

**🔹 What is Logging?**

Logging involves **recording detailed system and application events** (e.g., errors, warnings, info) that help developers and IT teams **trace, debug, and audit** behavior over time.

**🔍 Key Metrics to Monitor:**

| **Metric Type** | **Examples** |
| --- | --- |
| **System** | CPU, Memory, Disk I/O |
| **Application** | Request count, error rate, latency |
| **Infrastructure** | VM status, DB connections |
| **Custom/Business** | API usage, sales transactions |

**🔍 Types of Logs:**

| **Log Type** | **Used For** |
| --- | --- |
| Application | Code-level errors, exceptions |
| Server | OS, services, system events |
| Web | Access logs, request/response logs |
| Security | Auth attempts, role changes |
| Audit | Track user actions and config changes |

**🧰 3. Prerequisites**

* Application deployed (locally or to cloud)
* Basic understanding of your tech stack (e.g., .NET, Node.js)
* Access to a monitoring/logging tool

**🔧 4. Common Monitoring & Logging Tools**

| **Tool** | **Purpose** | **Platform** |
| --- | --- | --- |
| **Azure Monitor** | Full-stack cloud monitoring | Azure |
| **Application Insights** | Telemetry for .NET, JS apps | Azure |
| **Prometheus + Grafana** | System metrics & dashboards | Open-source |
| **ELK Stack (Elastic, Logstash, Kibana)** | Log collection/visualization | Open-source |
| **Datadog, New Relic** | Paid observability platforms | Cross-platform |

**🧪 5. Example: Monitoring with Azure Application Insights**

**✅ Step 1: Add Application Insights to ASP.NET Core app**

Install NuGet package:

dotnet add package Microsoft.ApplicationInsights.AspNetCore

Configure in Program.cs:

builder.Services.AddApplicationInsightsTelemetry();

**✅ Step 2: View Metrics**

* Go to Azure Portal → Your App Service → Application Insights
* View **Requests**, **Failures**, **Performance**, and **Live Metrics**

**📥 Sample Log Message in Code**

using Microsoft.ApplicationInsights;

TelemetryClient telemetry = new TelemetryClient();

telemetry.TrackEvent("UserRegistered");

telemetry.TrackException(ex);

**📸 6. Snapshot: Observability Workflow**

App Runs →

Logs Generated →

Sent to Monitoring Tool →

Dashboards / Alerts →

DevOps Teams Take Action

**✅ 7. Summary**

| **Key Takeaway** |
| --- |
| **Monitoring** helps you watch your app's health and performance in real-time, while **Logging** provides insights into what went wrong or what happened in the past. Both are essential for **high availability**, **performance tuning**, and **incident response** in DevOps. |