# Complete Observability System (Metrics, Logs & Traces)

## Introduction

This project sets up a complete observability stack for monitoring and troubleshooting microservices environments. It includes tools for metrics, logging, and tracing, essential for DevOps practices and cloud-native applications.

## Abstract

The observability stack integrates Prometheus for collecting metrics, Grafana for data visualization, Loki for centralized log aggregation, and Jaeger for distributed tracing. Using Docker Compose, this stack offers a seamless and portable environment to monitor applications, detect performance bottlenecks, and analyze system behavior in real-time.

## Tools Used

- Docker & Docker Compose
- Prometheus
- Grafana
- Loki
- Jaeger
- Sample Telemetry App
- Linux/Ubuntu (WSL2)

## Steps Involved in Building the Project

1. Defined the architecture and selected appropriate observability tools.
2. Configured Prometheus with scrape targets and alert rules.
3. Set up Grafana with pre-provisioned datasources and dashboards.
4. Configured Loki for collecting logs from Docker containers.
5. Integrated Jaeger for distributed tracing and visualizing spans.
6. Containerized all services and orchestrated them with Docker Compose.
7. Validated telemetry by generating logs, metrics, and traces using a test application.

## Conclusion

## Complete Observability System (Metrics, Logs & Traces)

This project successfully demonstrates how to build a full observability stack using open-source tools. It offers insights into system behavior, performance monitoring, and issue resolution. The system can be extended for production use with alerting, authentication, and persistent storage.