Faysal Aziz

Grafana Observability Engineering with Prometheus, Loki and OpenTelemetry

Building Unified Metrics, Logs and Telemetry Pipelines for Modern Cloud-Native Operations - 2 days

FA-0235 · 2 days

DevOps, Cloud & Infrastructure · Data & Analytics

Introduction

Modern infrastructure generates enormous volumes of operational data, yet many organizations still struggle to convert telemetry into actionable insight. Grafana, together with Prometheus, Loki, and OpenTelemetry, has become one of the most widely adopted open observability stacks for monitoring distributed systems, Kubernetes platforms, microservices, and cloud-native applications. The growing industry shift toward vendor-neutral telemetry standards and unified observability workflows has made OpenTelemetry and the Grafana ecosystem foundational technologies for DevOps, SRE, platform engineering, and operations teams. Recent industry trends also show increasing enterprise adoption of open observability platforms due to flexibility, scalability, and cost optimization advantages.

This intensive 2-day course is designed to provide participants with practical operational knowledge of Grafana dashboards, Prometheus metrics collection, Loki log aggregation, and OpenTelemetry instrumentation concepts without spending time on installation or environment setup procedures. The course focuses heavily on operational visibility, observability architecture, telemetry correlation, alerting strategies, query optimization, and real-world monitoring practices used in production environments. The instructor brings over 30 years of industry experience and emphasizes operationally relevant, production-oriented content aligned with current enterprise observability practices rather than purely academic coverage.

Learning outcomes

By the end of this course, participants will be able to:

- Understand modern observability architecture and telemetry workflows
- Differentiate between monitoring and observability concepts
- Explain the role of metrics, logs, traces, and telemetry pipelines
- Navigate and utilize Grafana for operational visualization and analysis
- Build and customize Grafana dashboards for infrastructure and application observability
- Understand Prometheus architecture, data models, and time-series concepts
- Write and optimize PromQL queries for operational monitoring
- Design effective alerting and notification strategies
- Analyze logs using Loki and LogQL
- Correlate logs, metrics, and traces within Grafana
- Understand OpenTelemetry architecture and telemetry signal generation
- Work with OpenTelemetry collectors, pipelines, and exporters conceptually
- Design scalable observability workflows for cloud-native environments
- Apply observability best practices for Kubernetes and distributed systems
- Understand operational challenges involving telemetry volume, retention, and scalability
- Implement observability governance and operational standards

Prerequisites

- Basic Linux command-line familiarity
- Understanding of networking fundamentals
- Basic knowledge of containers and Kubernetes concepts
- Familiarity with cloud-native application architectures
- Basic understanding of APIs and web services
- General operational or infrastructure administration experience is beneficial
- Prior exposure to monitoring tools is helpful but not mandatory

Training outline

1. Foundations of Modern Observability

Evolution from Traditional Monitoring to Observability

1. Monitoring versus observability
2. Operational visibility challenges
3. Reactive versus proactive operations
4. Distributed systems complexity
5. Cloud-native operational requirements

Core Observability Pillars

1. Metrics
2. Logs
3. Traces
4. Events
5. Telemetry correlation concepts

Open Observability Ecosystem

1. Grafana ecosystem overview
2. Prometheus role in metrics collection
3. Loki role in log aggregation
4. OpenTelemetry standardization concepts
5. CNCF observability landscape
6. Vendor-neutral observability strategies

Enterprise Observability Trends

1. OpenTelemetry adoption growth
2. Unified observability platforms
3. Cost-efficient telemetry architectures
4. AI-assisted observability trends
5. Observability in Kubernetes environments
6. Operational scalability considerations

2. Grafana Architecture and Operational Usage

Grafana Core Components

1. Grafana architecture overview
2. Data source integration concepts
3. Dashboard rendering workflows
4. User authentication and access models
5. Folder and organizational structures

Grafana Data Source Management

1. Prometheus data source integration
2. Loki data source integration
3. OpenTelemetry-compatible integrations
4. Multi-source observability workflows
5. Data source optimization considerations

Dashboard Design and Visualization

1. Dashboard structure and layout planning
2. Panels and visualization types
3. Time-series visualization strategies
4. Table and stat visualizations
5. Heatmaps and histogram concepts
6. Operational dashboard design standards

Variables and Dynamic Dashboards

1. Dashboard variables
2. Query-based variables
3. Dynamic filtering
4. Multi-environment dashboards
5. Template variable usage

Operational Analytics with Grafana

1. Correlating operational signals
2. Investigating service degradation
3. Infrastructure health visualization
4. Application performance visualization
5. Real-time operational analysis

Grafana Alerting and Notification Workflows

1. Unified alerting concepts
2. Alert rule creation
3. Threshold management
4. Multi-condition alerts
5. Alert routing strategies
6. Notification channels
7. Escalation workflows
8. Alert fatigue reduction practices

3. Prometheus Metrics Architecture and Operations

Prometheus Fundamentals

1. Prometheus architecture
2. Pull-based monitoring concepts
3. Time-series database fundamentals
4. Metrics lifecycle
5. Service discovery concepts

Prometheus Data Model

1. Metrics and labels
2. Time-series concepts
3. Dimensional monitoring
4. Cardinality considerations
5. Metric naming standards

Prometheus Metrics Types

1. Counters
2. Gauges
3. Histograms
4. Summaries
5. Native histogram concepts

PromQL Fundamentals

1. PromQL architecture
2. Instant queries
3. Range queries
4. Label filtering
5. Aggregation operators
6. Arithmetic operations
7. Rate calculations
8. Vector matching concepts

Advanced PromQL Operations

1. Recording rules
2. Query optimization
3. Forecasting and trend analysis
4. Percentile calculations
5. Histogram quantiles
6. Subqueries
7. Complex aggregation workflows

Prometheus Alerting Concepts

1. Alert rule architecture
2. Alert evaluation workflows
3. Alert state transitions
4. AlertManager concepts
5. Routing and silencing strategies
6. High-availability alerting considerations

Prometheus Operational Best Practices

1. Metric standardization
2. High-cardinality avoidance
3. Retention planning
4. Scaling Prometheus environments
5. Federation concepts
6. Long-term storage integration concepts

4. Loki Log Aggregation and Log Analytics

Loki Architecture Overview

1. Loki architecture fundamentals
2. Log indexing concepts
3. Label-based storage design
4. Distributed Loki architecture
5. Object storage integration concepts

Log Collection and Processing Concepts

1. Centralized logging strategies
2. Structured versus unstructured logs
3. Label extraction concepts
4. Log enrichment workflows
5. Multi-tenant logging considerations

LogQL Fundamentals

1. LogQL syntax
2. Log stream selection
3. Label filtering
4. Pattern matching
5. Parsing expressions
6. Aggregation operations

Advanced Loki Querying

1. Metrics from logs
2. Log-based alerting
3. Correlation workflows
4. Error pattern detection
5. Performance bottleneck investigation
6. Multi-source operational analysis

Loki Scalability and Retention

1. Storage optimization strategies
2. Query performance tuning
3. Retention policy planning
4. Large-scale log management
5. Cost optimization considerations

Operational Logging Best Practices

1. Logging standards
2. Structured logging practices
3. Context-aware logging
4. Correlation identifiers
5. Secure logging considerations
6. Compliance and audit logging

5. OpenTelemetry Architecture and Telemetry Pipelines

OpenTelemetry Fundamentals

1. OpenTelemetry architecture
2. Vendor-neutral observability
3. Telemetry signal concepts
4. Metrics, logs, and traces
5. Semantic conventions

OpenTelemetry Components

1. SDK concepts
2. API concepts
3. Collector architecture
4. Receivers
5. Processors
6. Exporters
7. Pipelines

Telemetry Collection Workflows

1. Instrumentation concepts
2. Auto-instrumentation strategies
3. Manual instrumentation concepts
4. Context propagation
5. Distributed tracing fundamentals

OpenTelemetry Metrics Concepts

1. Metric instruments
2. Synchronous metrics
3. Asynchronous metrics
4. Aggregation concepts
5. Temporality models

Distributed Tracing Concepts

1. Trace architecture
2. Spans and parent-child relationships
3. Context propagation workflows
4. Root cause analysis workflows
5. Service dependency visualization

OpenTelemetry Collector Operations

1. Pipeline design

2. Data transformation workflows
3. Filtering and sampling
4. Batch processing concepts
5. Export routing strategies
6. Reliability considerations

OpenTelemetry Integration with Grafana Ecosystem

1. Telemetry export workflows
2. Correlating metrics and logs
3. Trace-aware dashboards
4. Unified operational visibility
5. Cross-signal analytics

6. Unified Observability Operations

Correlating Metrics, Logs, and Traces

1. Cross-signal analysis
2. Incident investigation workflows
3. Root cause identification
4. Service dependency analysis
5. Telemetry-driven troubleshooting

Observability for Kubernetes and Cloud-Native Platforms*

1. Kubernetes observability challenges
2. Container telemetry concepts
3. Cluster operational visibility
4. Namespace and workload monitoring
5. Service mesh observability concepts

Displaimer

*This course outline is intended as a general training framework and guideline only. The instructor reserves the right to modify, reorganize, expand, condense, or otherwise amend the course content, sequence, coverage depth, demonstrations, and associated materials as deemed necessary to accommodate participant requirements, operational priorities, technological developments, audience proficiency levels, and time constraints without prior notice.

© 2026 Faysal Aziz · Kuala Lumpur