

Updated on 04/11/2025

Register

Grafana LGTM Stack Training

4 days (28 hours)

Overview

Our Grafana LGTM training course will enable you to master the comprehensive monitoring of a microservices architecture by integrating logs, metrics, and traces into a single observation space. You will learn how to deploy each component of the stack, create intelligent dashboards, explore complex traces, and configure correlated alerts.

This will enable you to identify incidents more quickly, reduce MTTR, trace inter-service calls, and anticipate performance drifts. You will also be able to integrate LGTM into an existing CI/CD pipeline or DevOps platform and adopt cloud-native observability best practices.

By the end of this training, you will be able to deploy, operate, and correlate monitoring data on the Grafana LGTM stack, using the latest stable versions of Grafana, Loki, Tempo, and Mimir.

Like all our training courses, this one is up to date with the latest [Grafana](#) updates.

Objectives

- Understand the pillars of observability (logs, metrics, traces)
- Know how to deploy and configure Grafana, Loki, Tempo, and Mimir
- Create interactive dashboards with data correlation
- Set up alerting, exploration, and advanced troubleshooting
- Integrate the LGTM stack into a DevOps pipeline or Kubernetes cluster

Target audience

- System administrators
- DevOps
- Cloud-native architects

- Backend developers

Prerequisites

- Basic knowledge of Linux (CLI, configuration files)
- Knowledge of Docker or Kubernetes appreciated
- Experience in DevOps, monitoring, or CI/CD recommended
- No prerequisites for Grafana or Prometheus

OUR GRAFANA LGTM TRAINING PROGRAM

Introduction to modern observability

- Definition of the pillars: logs, metrics, traces
- Differences between monitoring, logging, tracing, and alerting
- Cloud-native architecture: why observability is becoming critical
- Introducing the LGTM stack: Loki, Grafana, Tempo, Mimir
- Related standards: Prometheus, OpenTelemetry, OTLP
- DevOps integration and Kubernetes/cloud context

Getting started with Grafana

- Installing Grafana locally (Docker)
- Introduction to the interface: dashboards, panels, data sources
- Creating simple dashboards (CPU, memory, latency)
- Basic PromQL syntax in panels
- Importing/exporting JSON dashboards
- Workshop: Installing Grafana + creating a custom dashboard with Prometheus

Grafana Loki architecture and concepts

- Introduction to Loki: unindexed logs, label-based storage
- Collection with Promtail or Fluent Bit
- Querying with LogQL (filters, aggregations)
- Label vs. content: search optimization
- Integrating logs and metrics in the same dashboard
- Log structuring and label management

Working with Tempo for distributed traceability

- Concepts: trace, span, parent ID, trace ID
- Querying with TraceQL: filters, durations, services
- Integration with Grafana (Tempo data source)
- Instrumentation via OpenTelemetry (OTLP)

- Using the Tempo interface and correlated traces
- Workshop: Trace generation + visualization of distributed flows in Tempo

Mimir and long-term metrics

- Mimir positioning: scalable Prometheus backend
- TSDB format and Prometheus compatibility
- Creating alert rules in Mimir
- PromQL querying in Grafana (advanced)
- Multi-tenant management and namespaces
- Workshop: Deploying Mimir + Prometheus configuration? Mimir + visualization in Grafana

Correlating logs, metrics, and traces

- How to go from a metric to a trace
- Drilldown: logs? traces? root cause
- Building correlated dashboards
- Using variables, dynamic intervals
- Creating multi-data panels
- Use cases: 500 error, latency, saturation

Alerting and active monitoring

- Types of alerts: thresholds, absence, frequency, combined conditions
- Alerts on PromQL metrics
- Alerts on logs (specific patterns)
- Grafana alerting configuration (UI and files)
- Notification: Slack, email, PagerDuty
- Workshop: Creating multi-pillar alerts on HTTP errors and latency spikes

Production troubleshooting with LGTM

- Troubleshooting methodology: from symptom to cause
- Application failure simulation: comprehensive analysis
- Cross-analysis: errors in logs + metric spikes + long span
- Using Loki live tailing
- MTTR reduction with targeted exploration

Local LGTM deployment (Docker)

- Official or custom Docker Compose stack
- Configuration of volumes, ports, storage
- Integration of Prometheus, Tempo, Loki, and Mimir into a unified stack
- Test scenarios for validation
- Basic security (Grafana authentication, private network, etc.)

Deployment on Kubernetes

- Presentation of Grafana Labs Helm charts
- Installation of Grafana, Loki, and Tempo on Minikube or K3s
- Storage management (PVC, object storage, MinIO)
- Monitoring a K8s cluster via LGTM
- Best practices: persistence, logs, metrics, service traces

Best practices for observability design

- Structuring logs (JSON, relevant labels)
- Choosing relevant metrics (SLI/SLO)
- How to instrument your applications for traces
- OpenTelemetry standards and exporters
- Observability architectures GitOps / Infrastructure-as-code

Industrialization, auditing & scalability

- Integration with CI/CD (GitLab, Jenkins)
- Versioned dashboard templates
- Multi-environment monitoring (production, staging)
- Log, metric, and trace retention strategies
- Resource savings: log reduction, trace sampling, labels
- LGTM and observability roadmap for 2025+

Target companies

This training is intended for both individuals and companies, large or small, wishing to train their teams in a new advanced IT technology or to acquire specific business knowledge or modern methods.

Positioning at the start of training

The positioning at the start of the training complies with Qualiopi quality criteria. Upon final registration, the learner receives a self-assessment questionnaire that allows us to assess their estimated level of proficiency in different types of technologies, as well as their expectations and personal objectives for the upcoming training, within the limits imposed by the selected format. This questionnaire also allows us to anticipate certain connection or internal security issues within the company (intra-company or virtual classroom) that could be problematic for the monitoring and smooth running of the training session.

Teaching methods

Practical training: 60% practical, 40% theory. Training materials distributed in digital format to all participants.

Organization

The course alternates between theoretical input from the trainer, supported by examples and

discussion sessions and group work.

Assessment

At the end of the session, a multiple-choice questionnaire is used to verify that the skills have been correctly acquired.

Certification

A certificate will be issued to each trainee who has completed the entire training course.