Updated on 29/08/2025

Sign up

Grafana LGTM Stack training

4 days (28 hours)

Presentation

Our Grafana LGTM training course will enable you to master the complete supervision of a microservices architecture, integrating logs, metrics and traces in a single observation space. You'll 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'll also be able to integrate LGTM into an existing CI/CD pipeline or DevOps platform, and adopt cloud-native observability best practices.

On completion of this course, you will be able to deploy, exploit and correlate monitoring data on the Grafana LGTM stack, using the latest stable version of Grafana, Loki, Tempo and Mimir.

As with all our training courses, this one is up to date with the latest Grafana updates.

Objectives

- Understand the pillars of observability (logs, metrics, traces)
- Deploy and configure Grafana, Loki, Tempo and Mimir
- Create interactive dashboards with data correlation
- Implement 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, config files)
- Knowledge of Docker or Kubernetes appreciated
- Experience in DevOps, monitoring or CI/CD recommended
- No previous knowledge of Grafana or Prometheus necessary

OUR GRAFANA LGTM TRAINING PROGRAM

Introduction to modern observability

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

Getting started with Grafana

- Local installation of Grafana (Docker)
- Interface overview: dashboards, panels, data sources
- Creating simple dashboards (CPU, memory, latency)
- Basic PromQL syntax in panels
- JSON dashboard import/export
- Workshop: Installing Grafana + creating a custom dashboard with Prometheus

Grafana Loki architecture and concepts

- Introduction to Loki: non-indexed logs, label storage
- Collection with Promtail or Fluent Bit
- Query with LogQL (filters, aggregation)
- Label vs. content: search optimization
- Integration of logs and metrics in a single 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 datasource)
- Instrumentation via OpenTelemetry (OTLP)

- Using the Tempo interface and correlated traces
- Workshop: Generating traces + visualizing distributed flows in Tempo

Mimir and long-term metrics

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

Correlating logs, metrics and traces

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

Alerting and active supervision

- Alert types: 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-threaded alerts on HTTP errors and latency peaks

Troubleshooting in production with LGTM

- Troubleshooting methodology: from symptom to cause
- Application failure simulation: complete analysis
- Cross-analysis: errors in logs + peak metrics + long span
- Use of Loki live tailing
- MTTR reduction with targeted exploration

Local LGTM deployment (Docker)

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

Deployment on Kubernetes

- Presentation of Helm charts Grafana Labs
- 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 in observability design

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

Industrialization, audit & scalability

- CI/CD integration (GitLab, Jenkins)
- Versioned dashboard templates
- Multi-vendor supervision (prod, staging)
- Strategies for retaining logs, metrics and traces
- Resource saving: log reduction, sampling traces, labels
- LGTM roadmap and 2025+ observability

Companies concerned

This course is aimed at 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 on entry to training

Positioning at the start of training complies with Qualiopi quality criteria. As soon as registration is finalized, the learner receives a self-assessment questionnaire which enables us to assess his or her estimated level of proficiency in different types of technology, as well as his or her expectations and personal objectives for the forthcoming training course, within the limits imposed by the selected format. This questionnaire also enables us to anticipate any connection or security difficulties within the company (intra-company or virtual classroom) which could be problematic for the follow-up and smooth running of the training session.

Teaching methods

Practical course: 60% Practical, 40% Theory. Training material distributed in digital format to all participants.

Organization

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

and group work sessions.

Validation

At the end of the session, a multiple-choice questionnaire verifies the correct acquisition of skills.

Certification

A certificate will be awarded to each trainee who completes the training course.