

New

# OpenTelemetry - Mettre en oeuvre une architecture d'observabilité cloud-native

Instrumenter les applications, collecter les traces, métriques et logs, et structurer une chaîne d'observabilité ouverte

Présentiel ou en classe à distance



2 jours (14 h)

Forfait intra : 6.250,00 € HT

Réf.: SR391

La formation **OpenTelemetry, mettre en oeuvre une architecture d'observabilité cloud-native** permet d'acquérir une maîtrise concrète des pratiques modernes d'**observabilité** dans les architectures distribuées. Elle apporte les compétences indispensables pour **collecter, structurer et exploiter logs, traces et métriques** afin d'obtenir une vision unifiée du comportement des applications et des services cloud-native.

Pensée pour les environnements modernes orientés microservices, cette formation permet de comprendre la transition entre **supervision classique et observabilité**, en s'appuyant sur le standard **OpenTelemetry** et sur un backend complet de type **LGTM (Loki, Grafana, Tempo, Mimir)**. Les participants apprennent à instrumenter leurs applications, configurer un **OpenTelemetry Collector**, centraliser les signaux et construire des tableaux de bord corrélés pour accélérer l'analyse d'incidents, le diagnostic et l'amélioration continue des performances.

La formation met l'accent sur la mise en oeuvre opérationnelle des trois piliers de l'observabilité : **logs structurés avec Loki, traces distribuées avec Tempo, métriques applicatives avec Mimir / Prometheus**. Elle couvre également les bonnes pratiques indispensables en production : **labels, sampling, baggage, structuration des données, gestion de la cardinalité et sécurité des signaux**. Grâce à une approche très pratique, les participants développent une vision complète et exploitable de l'observabilité dans des environnements cloud et DevOps.

## A qui s'adresse cette formation ?



### Pour qui

- Développeurs
- SRE / DevOps
- Architectes applicatifs
- Ingénieurs observabilité / production
- Toute personne impliquée dans la supervision applicative moderne



### Prérequis

- Connaissances de base en développement applicatif (Java, .NET, Python ou équivalent)
- **Disposez-vous des connaissances nécessaires pour suivre cette formation ? Testez-vous !**

## Programme

### 1 - Passer de la supervision classique à l'observabilité cloud-native

- Comprendre les limites des outils de supervision IT traditionnels dans les architectures distribuées
  - Identifier les différences entre monitoring, supervision et observabilité cloud-native
  - Comprendre les bénéfices d'une approche orientée logs, traces et métriques corrélés
  - Analyser le rôle des trois signaux dans la détection proactive d'incidents
  - Découvrir les principaux outils d'observabilité modernes : OpenTelemetry, Grafana, Prometheus, Tempo, Loki
  - Introduire les enjeux du traçage distribué dans les architectures microservices
- Atelier

Analyser un incident applicatif et démontrer pourquoi la supervision classique ne suffit plus en environnement cloud-native

## 2 - Comprendre les fondamentaux d'OpenTelemetry

- Comprendre la genèse d'OpenTelemetry et sa standardisation par la CNCF
  - Identifier le rôle d'OpenTelemetry dans une stratégie d'observabilité unifiée
  - Distinguer OpenTelemetry API et OpenTelemetry SDK
  - Comprendre les trois signaux OpenTelemetry : logs, traces et métriques
  - Maîtriser les conventions sémantiques OpenTelemetry pour normaliser les données
  - Comparer les approches d'auto-instrumentation et d'instrumentation manuelle
- Atelier

Instrumenter un service applicatif simple afin de produire des traces distribuées, logs structurés et métriques locales

## 3 - Déployer OpenTelemetry Collector dans une architecture moderne

- Comprendre le rôle du OpenTelemetry Collector dans une architecture d'observabilité cloud-native
  - Structurer un pipeline de collecte : receivers, processors, exporters, pipelines
  - Utiliser les protocoles OTLP HTTP / gRPC pour la collecte de télémétrie
  - Appliquer les bonnes pratiques de configuration : batch, memory\_limiter, resource, attributes
  - Déployer OpenTelemetry Collector en mode agent, gateway, Docker ou Kubernetes
  - Optimiser la collecte des signaux dans des environnements distribués et hybrides
- Atelier

Installer un OpenTelemetry Collector et configurer un pipeline complet de collecte des logs, traces et métriques

## 4 - Centraliser et analyser les logs avec OpenTelemetry et Loki

- Comprendre le modèle de logs OpenTelemetry et sa structuration
  - Instrumenter des logs enrichis avec contexte, attributs et métadonnées
  - Configurer l'export OTLP vers Loki pour centraliser les logs applicatifs
  - Concevoir des labels efficaces et maîtriser la cardinalité des logs
  - Utiliser LogQL pour filtrer, parser, agréger et corréler les événements
  - Identifier rapidement erreurs, exceptions et signaux faibles dans les logs
- Atelier

Instrumenter les logs d'un microservice et analyser un incident applicatif via des requêtes LogQL dans Grafana

## 5 - Instrumenter et exploiter les traces distribuées avec Tempo

- Comprendre la propagation de contexte dans une architecture microservices
  - Maîtriser le modèle de traces OpenTelemetry : spans, parent/child, links, events
  - Configurer les stratégies d'échantillonnage : probabilistic sampling et tail-based sampling
  - Exporter les traces distribuées via OTLP vers Tempo
  - Visualiser les dépendances, latences et erreurs dans Grafana Tempo
  - Exploiter les traces pour diagnostiquer les goulets d'étranglement applicatifs
- Atelier

Instrumenter deux services communicants et analyser la chaîne complète d'exécution dans Tempo

## 6 - Produire et exploiter les métriques avec Mimir et Prometheus

- Comprendre les types de métriques OpenTelemetry : counter, gauge, histogram
  - Exporter les métriques applicatives vers Mimir et Prometheus
  - Utiliser PromQL pour interroger, agréger et corréler les métriques
  - Mesurer latence, disponibilité et performance avec quantiles et percentiles
  - Construire des indicateurs de supervision RED et USE
  - Exploiter les métriques pour piloter SLA, SLO et capacité applicative
- Atelier

Créer des métriques personnalisées, écrire des requêtes PromQL avancées et visualiser les résultats dans Grafana

## 7 - Corréler logs, traces et métriques dans Grafana

- Configurer les datasources Loki, Tempo, Mimir et Prometheus dans Grafana
- Construire un tableau de bord d'observabilité full-stack
- Mettre en oeuvre la méthode RED (Rate, Errors, Duration)
- Corréler logs, traces et métriques pour accélérer le diagnostic incident
- Créer des dashboards orientés performance, disponibilité et expérience utilisateur

- Définir un alerting simple avec PromQL et règles de seuil  
Atelier

Construire un dashboard "Health Service" corrélé et analyser un incident

## 8 - Maîtriser les concepts avancés d'observabilité

- Comprendre le rôle du Baggage et sa propagation
- Structurer les attributs standard et personnalisés
- Maîtriser la volumétrie (sampling, anti-cardinalité)
- Sécuriser les signaux : masquage, données sensibles, conformité
- Définir une stratégie d'observabilité adaptée à la production  
Atelier

Ajouter un baggage métier et visualiser sa propagation dans une trace



### Les objectifs de la formation

- Comprendre les différences entre supervision traditionnelle, monitoring applicatif et observabilité cloud-native
- Maîtriser les concepts fondamentaux d'OpenTelemetry : API, SDK, Collector, signaux et conventions sémantiques
- Instrumenter une application pour produire des logs structurés, traces distribuées et métriques exploitables
- Déployer et configurer un OpenTelemetry Collector pour centraliser la collecte des données de télémétrie
- Exporter les logs vers Loki, les traces vers Tempo et les métriques vers Mimir / Prometheus
- Construire des tableaux de bord Grafana corrélant logs, traces et métriques pour accélérer le diagnostic
- Exploiter PromQL et LogQL pour analyser la performance, les erreurs et les incidents applicatifs



### Evaluation

- Pendant la formation, le formateur évalue la progression pédagogique des participants via des QCM, des mises en situation et des travaux pratiques. Les participants passent un test de positionnement avant et après la formation pour valider leurs compétences acquises.



### Les points forts de la formation

- Un cas fil rouge réaliste, basé sur une application multi-services instrumentée de bout en bout
- Une approche opérationnelle, avec ateliers d'instrumentation, corrélation et analyse d'incidents
- Une vision unifiée de l'observabilité, pour corréler efficacement logs, traces et métriques dans Grafana
- Des bonnes pratiques directement applicables en environnement DevOps / SRE, pour améliorer performance, fiabilité et résilience