Schulte

Exposé zum Konzept

Konzeption und Implementierung von Observability mittels OpenTelemetry

Vorgelegt bei: loremipsum℠

Erstellt von: Thomas-Laurenz Schulte
thoschulte@gmail.com

Inhaltsverzeichnis

1 Motivation

In der heutigen schnelllebigen und komplexen Welt der Cloud-Technologie ist es essenziell, Echtzeit-Einblicke in die Leistung und das Verhalten von Anwendungen zu haben.

Observability ist nicht mehr nur ein nettes Extra, sondern eine zentrale Anforderung für die Entwicklung und den Betrieb zuverlässiger, skalierbarer und sicherer (Cloudnative) Systeme. Es ermöglicht nicht nur zu sehen, was in einem System passiert, sondern auch zu verstehen, warum etwas passiert. Dies ist entscheidend, um proaktiv auf Probleme zu reagieren, die Benutzererfahrung zu verbessern und letztendlich das Produkt kontinuierlich zu optimieren.

Die Wahl von OpenTelemetry als Instrumentierungstool für Observability in diesem Projekt ist durch mehrere Faktoren motiviert. OpenTelemetry bietet einen einheitlichen und herstellerneutralen Ansatz, um Telemetriedaten zu sammeln, was es zu einem vielseitigen Werkzeug für die Überwachung verschiedenster Anwendungen macht. Es ist ein CNCF-Projekt (Cloud Native Computing Foundation) und genießt breite Unterstützung durch die Open-Source-Gemeinschaft, was kontinuierliche Verbesserungen und eine hohe Adaptionsrate verspricht. Darüber hinaus ist OpenTelemetry speziell darauf ausgelegt, nahtlos mit Cloudnative Technologien zu arbeiten und fördert Best Practices in der Softwareentwicklung und -wartung.

2 Zielsetzung der Arbeit

2.1 Hintergrund

Das geplante Konzept beschäftigt sich mit der Implementierung von Observability in eine Anwendungslandschaft unter Verwendung von OpenTelemetry. Observability, ein entscheidender Aspekt moderner Softwareentwicklung, ermöglicht es, Einblicke in die Leistung, Zuverlässigkeit und das Verhalten von Anwendungen zu erhalten. (Cloudnative) Anwendungen, die auf Microservices, dynamischer Orchestrierung und kontinuierlicher Integration, respektive Deployment basieren, stellen besondere Herausforderungen an die Überwachung und das Performance-Management.

OpenTelemetry, ein Satz von APIs, Bibliotheken und anderen Komponenten, die zur Beobachtung von Anwendungen genutzt werden, bietet Werkzeuge zur Erfassung, Verarbeitung und Exportierung von Telemetriedaten. Es handelt sich hierbei um ein Open-Source-Projekt, das von der Cloud Native Computing Foundation unterstützt wird und darauf abzielt, einen einheitlichen und standardisierten Weg zur Sammlung und Übertragung von Telemetriedaten wie Traces, Metriken und Logs bereitzustellen. OpenTelemetry entstand aus der Zusammenführung zweier separater Open-Source-Projekte, OpenTracing und OpenCensus.

2.2 Ziele

Ziel der Arbeit ist, ein Konzept zur Integration von OpenTelemetry ins Frontend und Backend zu erarbeiten und dieses anhand einer Implementierung in eine Cloudnative Anwendung mit dem Express Framework und einer Angular 17 Anwendung als Proof of Concept zu verdeutlichen. Mithin soll die Verbesserung der Observability aufgezeigt werden. Observability bezieht sich auf die Fähigkeit eines Systems, Einblicke in sein Verhalten und seine Leistung zu gewähren, basierend auf den externen Ausgaben, die es produziert, insbesondere in Form von Logs, Metriken und Traces. Ferner soll aufgezeigt werden, dass damit einhergehende Monitoring Anwendungen zugutekommt und Teams dabei unterstützt, Fehler in der Anwendung schnell zu finden, einen möglichen Ausfall zeitnah beheben oder im Vorfeld erkennen zu können.

Spezifischen Ziele:

1. Transparenz: Ein umfassendes Verständnis dafür gewinnen, wie die Anwendung unter verschiedenen Bedingungen funktioniert.
2. Fehlererkennung und Diagnose: Schnelle Identifizierung und Behebung von Problemen innerhalb der Anwendung, was zu einer verbesserten Zuverlässigkeit und Benutzerzufriedenheit führt.
3. Leistungsüberwachung: Verständnis der Anwendungsperformance in Echtzeit, um Engpässe zu identifizieren und zu beheben.
4. Verhaltensanalyse: Verfolgung des Benutzerverhaltens und der Interaktionen mit der Anwendung, um die Nutzererfahrung zu verbessern.
5. Verteiltes Tracing: Verfolgung von Anfragen, die durch verschiedene Services und Komponenten einer Microservices-Architektur laufen, um ein vollständiges Bild der Anfragerouten zu erhalten.
6. Skalierbarkeitsmanagement: Messen der Auswirkungen von Skalierungs-operationen auf die Systemleistung, um fundierte Entscheidungen über die Ressourcennutzung zu treffen.
7. Entwicklungs- und Betriebseffizienz: Bietet Entwicklern und Betriebsteams Daten, die für die schnelle und effiziente Lösung von Problemen und die Verbesserung der Systeme notwendig sind.
8. Kostenoptimierung: Analyse von Telemetriedaten zur Identifizierung von Kostentreibern und zur Optimierung der Ressourcennutzung.

Durch die Einbettung von OpenTelemetry in Anwendungen sollen Teams also nicht nur Probleme schneller lösen, sondern auch proaktiv Systeme optimieren und die allgemeine Qualität und Stabilität ihrer Softwarelösungen verbessern können.

Überdies soll nach erfolgreicher Konzeptions- und Implementierungsphase die Arbeit in ein Minimum Viable Product (MVP) überführt werden können.

2.3 Abgrenzung

Das geplante Konzept konzentriert sich speziell auf die Konzeption und Implementierung von Observability-Strategien unter Verwendung von OpenTelemetry. Während das breite Feld der Observability zahlreiche Aspekte umfasst, wie z.B. umfassendes Monitoring, umfangreiche Log-Analyse und vollständige Infrastruktur-Überwachung, wird dieses Projekt sich gezielt mit den Kernfunktionen, mithin Konzeption und Konstruktion von OpenTelemetry auseinandersetzen. Es werden dabei die grundlegenden Mechanismen für Tracing, Metriken und Logging innerhalb einer Anwendungslandschaft erforscht und implementiert. Andere Aspekte der Observability, die über den Rahmen von OpenTelemetry hinausgehen oder spezifische Third-Party-Tools erfordern, liegen außerhalb des Umfangs dieser Arbeit. Ebenso werden Implementierungsdetails, die für spezifische Cloud-Anbieter oder Plattformspezifische Services gelten, nicht im Detail betrachtet, um die Allgemeingültigkeit und Übertragbarkeit der Ergebnisse zu gewährleisten.

3 Gegenstand der Arbeit

3.1 Vorgehen

Gegenstand der Arbeit wird zunächst nach anfänglicher Recherche eine ausführliche Konzeption der IT-Infrastruktur für OpenTelemetry, der Backend- und Frontend-Services, mithin deren entsprechender Instrumentalisierung sein. Dazu zählt nach sorgfältiger Analyse diverser Aspekte wie Kommunikationsfluss, Performance-Baselines und Log-Strategien etc. als erstes die Erstellung eines passenden Grobkonzepts, das als Entscheidungsgrundlage auf hoher Ebene dient, während das darauffolgende Feinkonzept für die tatsächliche Umsetzung und das detaillierte Projektmanagement definiert.

Im Anschluss an die Konzeptionsphase, folgt die Implementierung des geplanten Systems. Dies umfasst die Entwicklung, Programmierung, Konfiguration und Einrichtung aller Systemkomponenten:

- Entwicklung IT-Infrastruktur
  - docker-compose.yml
- Konfiguration Grafana
  - datasources.yml
- Konfiguartion Prometheus
  - prometheus.yml
- Konfiguration OpenTelemetry Collector
  - otel-config.yml
- Programmierung Backend-Services
  - node.js Express-Framework
- Programmierung des Frontend-Service
  - Markup / HTML

    - Web-Server, Template-Engine
- Entwicklung Dockerfiles - Dockerisierung der Services
- Instrumentalisierung

3.2 Erwartungen

Am Ende dieses Projekts wird erwartet, eine umfassende Lösung für Observability etabliert zu haben, die auf OpenTelemetry basiert und in die Anwendungsarchitektur integriert ist.

Konkret sollen folgende Resultate erzielt werden:

1. Implementierte Observability-Funktionen: Ein vollständig integriertes Observability-System, das in der Lage ist, Traces, Metriken und Logs effektiv zu erfassen, zu verarbeiten und darzustellen.
2. Nachweisbare Systemverbesserungen: Messbare Verbesserungen in der Überwachung der Anwendungsperformance, der Fehlererkennung und der Betriebseffizienz, die durch die Einführung von OpenTelemetry erreicht werden.
3. Entwicklungsdokumentation: Eine detaillierte Dokumentation der Konzeptions- und Implementierungsprozesse, die als Referenz für zukünftige Implementierungen von Observability in ähnlichen Projekten dienen kann.
4. Leistungsbericht: Eine Evaluation der Leistungsfähigkeit von OpenTelemetry in der prototypischen Anwendung, einschließlich einer Analyse von Latenzzeiten, Datendurchsatz und der Genauigkeit der Telemetriedaten.
5. Einsatzempfehlungen: Praxisnahe Empfehlungen für die Skalierung und Optimierung der Observability-Strategie für Cloud-native Anwendungen, die auf den gewonnenen Erkenntnissen basieren.

Durch die Erreichung dieser Ergebnisse wird das Projekt nicht nur die technischen Aspekte von Observability adressieren, sondern auch einen wertvollen Beitrag zur Best-Practice-Entwicklung in diesem Bereich leisten und so überdies die Auffassung fördern, dass die Implementierung von OpenTelemetry effektiv zur Verbesserung der Qualität einer Software beitragen und technische Schulden vermeiden kann.

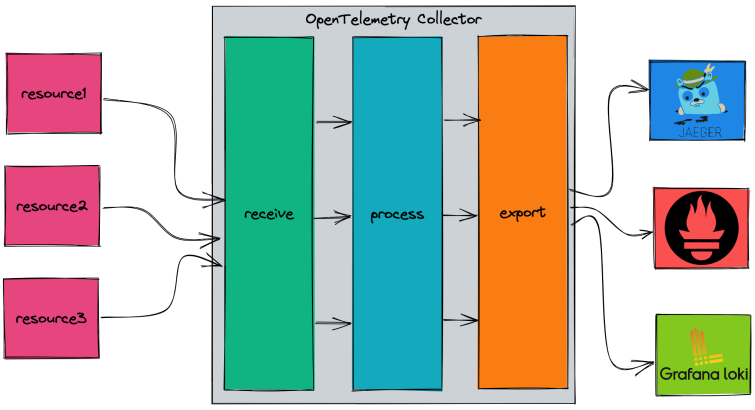

Abbildung 1: Erwartete Architektur

4 Vorläufige Gliederung

Licence
Vorwort
Abstract
Inhaltsverzeichnis
1. Einleitung
    1.1 Motivation
    1.2 Ziel der Arbeit
    1.3 Aufbau der Arbeit
2. Grundlagen
    2.1 Cloudnative
    2.2 Microservices
    2.3 Observability
    2.4 OpenTelemetry
3. Technologien
    3.1 Traces
        3.1.2 Jaeger
        3.2.3 Zipkin
    3.2 Metrics
        3.2.1 Prometheus
        3.3.2 Datadog
    3.3 Logs
        3.3.1 Grafana Loki, Promtail
        3.3.2 Logstash
        3.3.3 Fluentd
    3.4 Wahl der Softwarekette
4. Anforderungen
    4.1 Funktionale Anforderungen

5 Literatur

Charity Majors, Liz Fong-Jones, George Miranda (2022). Observability Engineering. O'Reilly Media

Manisha Agrawal, Karun Krishnannair (2023). Implementing Enterprise Observability for Success: Strategically Plan and Implement Observability Using Real-life Examples. Packt Publishing

Cornelia Davis (2019). Cloud Native Patterns: Designing Change-tolerant Software. Manning

Tom Laszewski, Kamal Arora, Erik Farr, Piyum Zonooz (2018). Cloud Native Architectures: Design High-availability and Cost-effective Applications for the Cloud. Packt Publishing

Alex Boten, Charity Majors (2022). Cloud-Native Observability with OpenTelemetry: Learn to Gain Visibility Into Systems by Combining Tracing, Metrics, and Logging with OpenTelemetry. Packt Publishing

Manuel Spigolon, Maksim Sinik, Matteo Collina (2023). Accelerating Server-Side Development with Fastify: A Comprehensive Guide to API Development for Building a Scalable Backend for Your Web Apps. Packt Publishing