쿠버네티스 환경에서 오픈텔레메트리 수집기를 적용한 관측 가능성 시스템 설계

차동헌, 김영한*
숭실대학교

dcha94@dcn.ssu.ac.kr, *younghak@ssu.ac.kr

Design of Observability System with OpenTelemetry Collector in Kubernetes

Cha Dong Hun, Kim Young Han*
Soongsil Univ.

요 약

현재 최신 기술의 발달과 함께 그에 맞는 유연하고 효율적인 서비스를 위하여 기존 모놀리식 방식의 아키텍처에서 벗어나 컨테이너 중심의 클라우드 네이티브 인프라 구축이 대두되고 있다. 따라 클라우드 네이티브 인프라 구축을 위한 대표적인 오케스트레이션 플랫폼인 쿠버네티스를 포함하여 운영 전반적에 걸쳐서 다방면으로 연구가 진행중이다. 특히 클라우드 네이티브 인프라에서 안정성 및 운영을 위해 관측 가능성의 필요성을 제시하였고, 이에 포함되는 메트릭, 로그, 분산 추적 등이 활용되고 있다. 본 논문에서는 대표적인 클라우드 네이티브 환경인 쿠버네티스에서의 서비스에 오픈텔레메트리 수집기를 적용하여 관측 가능성에 대한 요소들을 제공하는 시스템 모델을 제시한다.

Ⅰ. 서 론

현재 IoT, 인공지능 등의 첨단 기술로 발달로 인한 데이터 전송량 및 요구량이 급증함에 따라 서비스를 제공하기 위해 구성되는 인프라에도 변화가 일어나고 있다. 많은 기업 및 서비스 제공 업체들은 소비자의 요구에 맞게 클라우드 네이티브(Cloud Native) 인프라를 구축하여 확장성 및 관리 편의성, 안정성을 추구하고 있다.

하지만 이런 확장성이 장점인 만큼, 그만큼 클라우드 인프라의 복잡도 증가는 필연적일 수밖에 없다. 이런 복잡해지는 애플리케이션과 클라우드 네이티브 인프라로 인한 모호함의 축소를 위해 서비스들의 관계 정립 및 관리를 위한 관측 가능성(Observability) 확보에도 연구가 활발히 진행되고 있다. 개별 리소스의 동작 및 상태 모니터링을 넘어 시스템 상태를 전체적으로 모니터하는 의미의 관측 가능성 요소로는 대표적으로 정량적 수치로 표현되는 메트릭(Metric)이 존재하며 그 외에도 타임스탬프 별로 발생되는 이벤트의 기록인 로그(Log)와, 종단 간 요청의 흐름을 표현하기 위한 추적(Trace) 등이 존재한다. 이런 관측 가능성 데이터들을 이용하여 전체적인 시스템의 시각화와 그 기반으로 오류를 탐색하는 것이 클라우드 네이티브 환경에서 지속적인 프로세스의 제공과 인프라 내 문제 방지의 대표적 방법으로 꼽힌다.

따라서 본 논문에서는 대표적인 클라우드 네이티브 환경인 Kubernetes[1]에서 서비스를 제공할 때, OpenTelemetry 수집기를 적용하여 가능성 요소인 추적 데이터와 메트릭을 동시 제공하고 그 데이터를 바탕으로 오류를 탐지해 낼 수 있는 시스템 모델을 제안한다.

Ⅱ. 관련 연구

OpenTelemetry(OTel, 오픈텔레메트리)는 클라우드 네이티브 환경에서 관측 가능성의 요소들로 여겨지는 메트릭, 로그 와 분산 추적 등의 생성 및 관리를 위해 설계된 API, SDK 및 도구와 표준화 등을 관리하는 프로젝트이다.[2] 이는 CNCF(Cloud Native Computing Foundation)에서 분산 추적 및 컨텍스트 전파를 위해 일관된 업체 중립 API 를 제공하던 프로젝트들인 OpenTracing[3] 및 OpenCencus[4] 가 병합되어 창설되었다. OpenTelemetry 에서는 클라우드 네이티브 환경에서의 관측 프레임 워크를 제공하기 위해 관련 계측 라이브러리를 제공하고 있으며 공급 업체에 구애받지 않는 계측 데이터를 수집하고 내보낼 수 있는 수집기, 측정 데이터를 전송하고 수신하기 친화적인 프로토콜인 OTLP (OpenTelemetry Protocol) 또한 정의하고 있다.

수집기는 Receiver, Processor, Exporter 로 구현되며 해당하는 라이브러리와 함께 관측 가능성 요소들을 수신하고 처리, 특정 백엔드나 다른 모듈로 내보내는 역할을 맡는다.

그림 1. 오픈텔레메트리 수집기 구성

Ⅲ. 본 론

본 논문에서는 마이크로 서비스로 이루어진 어플리케이션를 배포한 쿠버네티스 환경에 OpenTelemetry 수집기를 적용하여 발생하는 추적 데이터와 메트릭을 동시 수집하고 제공하는 시스템을 구성하였다.

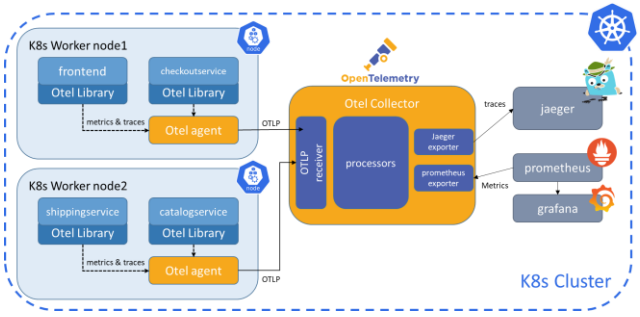

그림 2. 쿠버네티스 관측 가능성 시스템 아키텍처

위 그림에서 구성한 쿠버네티스에서 사용한 온라인 쇼핑몰 어플리케이션은 여러 마이크로 서비스로 구성되어 있고, 일관되지 않은 언어로 작성되어 있다.[5] 이 어플리케이션을 구성하는 마이크로 서비스에 각각 OpenTelemetry 에서 제공하는 언어별 계측 라이브러리를 추가하여 요청 및 응답 프로토콜에 고유 식별자를 추가할 수 있게 구성한 뒤 배포하였다. 그 외에도 노드 별 관측 가능성 요소 데이터를 수집하기 위한 daemonset 으로 구성된 OpenTelemetry agent 를 배포하였고, Opentelemetry 수집기와 데이터를 시각화 하기 위한 Jaeger, Prometheus 등을 배포하였다.

이후 쿠버네티스 클러스터 안에서 발생하는 관측 가능성 요소인 추적 및 메트릭은 발생 시 계측 라이브러리로 인하여 각 노드에 구성된 OpenTelemetry agent 로 전송된다. 전송된 데이터는 OTLP 를 이용해 쿠버네티스 클러스터 안에 Opentelemetry 수집기로 재전송되며, 이런 데이터들은 수집기 내에서 프로세싱을 거쳐 원하는 백엔드로 보내는 것이 가능하다. 추적 데이터의 경우 분산 추적 오픈소스 프로젝트인 jaeger 로 전송하여 마이크로 서비스의 그래프화 및 추적 데이터 시각화를 담당하였고, 메트릭의 경우 prometheus_exporter 로 노출되며 이는 prometheus 로 수집되어 제공될 것이다.

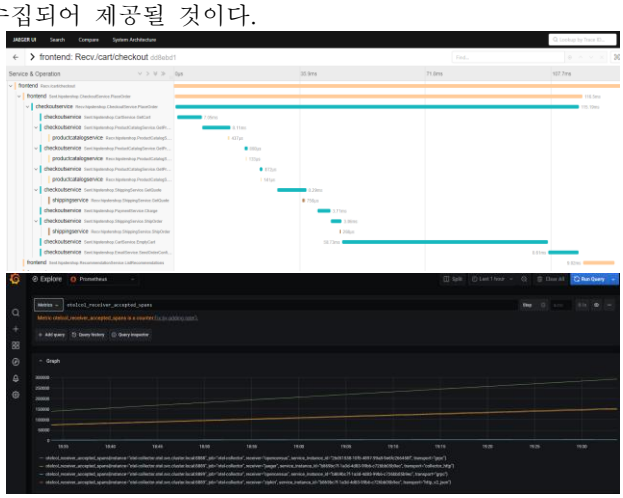

그림 3. 계측된 데이터의 제공 화면 예시

서비스 요청에 따라 생성된 추적 데이터는 요청 발생 시 생성된 요청과 통신한 모든 마이크로 서비스를 타임스탬프에 맞춰 표시하며, 전송 방향이 표시된 그래프로도 표현이 가능하다. 또한 추적 데이터와 더불어 요청의 수나 응답시간 등의 메트릭 또한 확인이 가능해 문제 발생 시 코드 레벨의 수정 등 빠른 대응이 가능할 것으로 보인다.

Ⅳ. 결 론

본 논문에서는 쿠버네티스 환경에서 원활한 서비스를 위해 OpenTelemetry 수집기를 적용한 관측 가능성 요소의 동시 제공 시스템 모델을 제안하였다. 제공되는 메트릭과 추적 데이터를 통하여 배포된 서비스의 전체적인 그래프화와 서비스의 품질 또한 함께 측정할 수 있을 것으로 보이며, 상호 비교와 검토를 통해 오류 검출과 전체적인 인프라 구성 인지 및 수정에 도움이 될 것으로 보인다. 하지만 이러한 환경 구성에는 서비스에 대응하는 라이브러리를 직접 추가하여야 하는 단점을 가지며 이후 연구를 통하여 이를 보완할 수 있는 방법을 제공할 계획이다.

ACKNOWLEDGMENT

"이 논문은 2021 년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No.2020-0-00946, 하이브리드 클라우드 환경에서의 고속, 자동 서비스 복구 및 이전 소프트웨어 개발) "

참 고 문 헌

[1] Kubernetes, “Kubernetes”, 2021, (https://kubernetes.io/)

[2] OpenTelemetry, Observability framework for cloud-native, 2021, (https://opentelemetry.io/)

[3] OpenTracing, Vendor-neutral APIs for distributed tracing, 2021, (https://opentracing.io//)

[4] OpensCensus, Library for metrics and distributed traces, 2021, (https://opencensus.io/)

[5] microservice-demo, Online Boutique, 2021, (https://github.com/GoogleCloudPlatform/microservices-demo)