쿠버네티스 환경에서의 로그 및 이벤트 관리 Operator 기반 설계

주형우, 손성혜, 김영한*
*숭실대학교

bono1130z@dcn.ssu.ac.kr, wnguddn1234@dcn.ssu.ac.kr, *younghak@ssu.ac.kr

Operator-Based Design for Log and Event Management in Kubernetes Environments

Ju Hyeong Woo, Son Sung Hye, Kim Young Han*
*Soongsil Univ.

요 약

클라우드 환경에서 모니터링을 위한 다양한 오픈소스 프로젝트가 존재한다. 본 논문에서는 쿠버네티스 네이티브한 솔루션을 통해 로그와 이벤트를 효율적으로 수집, 관리 및 시각화하는 CRD(Custom Resource Definition) 및 Operator 기반의 접근 방식을 제안한다. 이를 통해 문제 발생 시 신속하게 원인을 분석하고 이에 대한 대응을 가능하게 하여 시스템 운영의 효율을 향상시킬 수 있을 것으로 기대된다.

Ⅰ. 서 론

대규모 클라우드 환경에서 문제가 발생했을 때, 발생 지점을 파악하고 문제를 해결하는 것은 매우 중요하다. 이러한 환경에서 문제를 신속하게 발견하고 해결하기 위한 다양한 모니터링 오픈소스 프로젝트가 존재하지만 많은 수의 컨테이너와 마이크로 서비스를 사용하는 환경에서 로그 및 이벤트를 관리하는 것은 여전히 어려운 문제이다.[1]

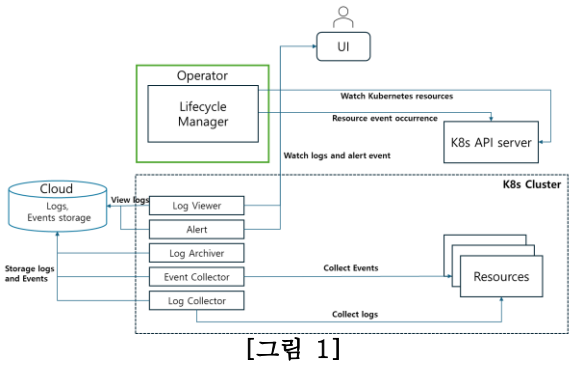
쿠버네티스 환경은 다수의 네임스페이스와 마이크로 서비스로 구성되어 있어, 각 서비스의 로그와 이벤트를 효과적으로 관리하는 것이 시스템 운영의 핵심 요소이다. 이를 통해 문제가 발생했을 때 빠르게 원인을 분석하고 대응할 수 있다.

기존의 로그 수집 방법은 주로 모니터링 오픈소스 프로젝트인 Fluentd[2], Elasticsearch[3], Kibana[4] 등을 사용하지만, 이러한 방법은 설정과 유지보수에 많은 비용이 들고, 사용자의 환경에 맞춰 변형하는 데 한계가 있다. Fluentd 는 다양한 플러그인을 통해 여러 데이터 타입의 로그를 수집할 수 있으나 초기에 구성하는 데 많은 시간이 소요된다. Elasticsearch 는 로그 데이터를 인덱싱하여 검색하는 데 특화되어 있으나 운영자의 역량에 따라 소요되는 리소스가 크게 차이날 수 있다. Kibana 는 로그 데이터를 시각화하는 데 유용하나, 로그 데이터를 시각화 하는 데 집중되어 있어 로그를 관리하는 부분이 부족하다.

따라서, 본 논문에서는 쿠버네티스 네이티브 솔루션을 통해 로그와 이벤트를 효율적으로 수집, 관리 및 시각화하기 위한 CRD 와 Operator 를 제안한다.[5][6] 로그 및 이벤트를 중앙 저장소로 수집하고, 특정 로그 패턴이나 이벤트를 기반으로 사용자에게 경고 메시지를 전송한다. 또한 이를 사용자 인터페이스(UI)로 시각화하여 사용자의 편의성을 증가시키고 로그와 이벤트를 통합된 방식으로 관리할 수 있도록 한다.

본론에서는 LogCollector, EventCollector, LogViewer, AlertRule, LogArchiver 의 다섯 가지 CRD 를 통해 로그와 이벤트를 관리하는 방법을 설명하고 CRD 가 동작하기 위한 Operator 구조를 제안한다. 이를 통해 로그 및 이벤트 데이터에 쉽게 접근하고 문제 발생 시 신속하게 대응할 수 있게 되어 시스템 운영의 효율이 향상될 수 있을 것으로 기대된다.[7]

Ⅱ. CRD 및 Operator 구조 설계



[그림 1]

[그림 1]은 리소스의 로그와 이벤트 데이터가 정의된 CRD 와 Controller 에 의해 중앙 로그 저장소에 저장되고 사용자가 CRD 로 설정한 대로 Controller 가 사용자에게 데이터를 시각화하여 보여주는 과정이다.

먼저 LogCollector 와 EventCollector 를 통해 리소스들의 로그 데이터와 이벤트 데이터를 중앙 로그 저장소로 수집한다. LogCollector 와 EventCollector 는 각 네임스페이스에서 설정되며, 지정된 로그와 이벤트

소스로부터 주기적으로 로그를 읽어와 중앙 저장소에 저장한다.

LogCollector 의 Controller 는 각 Pod 의 컨테이너 로그 파일을 주기적으로 읽어와 중앙 로그 저장소로 전송한다. 이는 컨테이너 런타임의 로그 파일을 직접 읽어오는 방식으로 이루어진다. EventCollector 의 Controller 는 모든 쿠버네티스의 이벤트를 읽어온다. 쿠버네티스 이벤트는 events 리소스로 관리되므로, 이를 통해 발생하는 이벤트를 실시간으로 수집한다.


```
{
    "logMessage": "Error: Failed to connect to database",
    "timestamp": "2024-05-14T10:15:30Z",
    "namespace": "production",
    "serviceName": "user-service",
    "podName": "user-service-5678d4f6b7-abcde",
    "containerName": "user-container"
}

{
    "eventMessage": "Pod user-service-5678d4f6b7-abcde is in CrashLoopBackOff",
    "timestamp": "2024-05-14T10:17:45Z",
    "namespace": "production",
    "serviceName": "user-service",
    "eventType": "Warning"
}
```

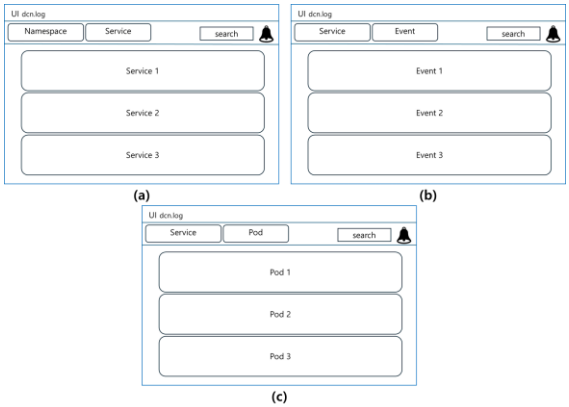

[그림 2]

[그림 2]와 같이 로그 파일은 JSON 형식으로 구조화되어 이후 검색 및 분석이 용이하도록 한다. 로그 데이터는 로그 메시지, 시간, 네임스페이스, 서비스 이름, Pod 이름, 컨테이너 이름 등의 필드를 포함하고 있으며 이벤트 데이터는 이벤트 메시지, 시간, 네임스페이스, 서비스 이름, 이벤트 유형 등의 필드를 포함하도록 한다.

이런 데이터들이 중앙 저장소에 저장되는 것을 관리하는 CRD 가 LogArchiver 이다. 사용자가 데이터 보존 기간을 특정하는 필드가 있다. LogArchiver Controller 가 설정된 주기마다 중앙 로그 저장소를 검사하고, 설정된 보존 기간이 지난 데이터를 삭제한다.

문제가 발생했을 때 사용자에게 알려주기 위해 특정 로그 및 이벤트를 기반으로 경고 알람을 설정한다. 이는 AlertRule 을 통해 구현한다. 사용자는 알람을 받기 원하는 로그 오류나 이벤트를 지정하고 횟수를 설정한다. AlertRule Controller 는 설정된 알림 규칙에 따라 동작한다. 중앙 로그 저장소를 지속적으로 모니터링하여 사용자가 지정한 키워드를 탐지한다. 키워드가 지정된 횟수만큼 발생하면 사용자에게 경고 메시지를 전송한다

특정 네임스페이스나 서비스를 쉽게 조회할 수 있도록 LogViewer 를 통해 사용자에게 제공하는 설정을 관리한다. 로그 및 이벤트는 네임스페이스, 서비스, 시간 범위 등으로 필터링되며, 이를 바탕으로 검색이 가능하다. 또한 로그 데이터를 직접 찾지 않고 이벤트 ID 를 기준으로 관련된 로그 데이터를 모두 조회할 수 있도록 데이터 구조를 설계한다. LogViewer Controller 는 사용자의 설정을 기반으로 하여 로그 및 이벤트 데이터를 필터링하고 이를 사용자에게 제공한다.



[그림 3]

[그림 3]는 사용자 인터페이스 예시이다. (a)는 인터페이스의 첫 화면이다. AlertRule 을 통해 사용자에게 경고 알람이 오면 오른쪽 위의 알람 아이콘을 통해 확인할 수 있다. 왼쪽 위의 네임스페이스를 통해 네임스페이스를 선택하고, 별로 서비스를 선택할 수 있다. 검색창을 통해 키워드를 검색할 수 있다.

서비스를 선택하면 (b)와 같은 인터페이스가 표시된다. 선택된 서비스의 이벤트가 시간 순서대로 정렬되며, 로그를 확인하고 싶은 이벤트를 클릭하면 해당 이벤트와 연관된 로그를 표시한다.

특정 Pod 의 모든 로그를 확인하고 싶을 경우 Event 를 Pod 로 변경하면 (c)와 같은 형태의 인터페이스가 나타나고 Pod 를 선택해 해당 Pod 의 모든 로그를 확인할 수 있다.

Ⅲ. 결론

본 논문에서는 쿠버네티스 환경에서 로그 및 이벤트 데이터를 효율적으로 수집, 관리, 저장 및 조회하기 위한 CRD 및 오퍼레이터 기반 솔루션을 제안했다. 제안된 시스템은 LogCollector 와 EventCollector 를 통해 로그와 이벤트 데이터를 중앙 로그 저장소로 수집하고, LogArchiver 를 통해 데이터 보존 기간을 설정한다. 또한 AlertRule 로 특정 로그 및 이벤트에 대한 알람을 설정하고 LogViewer 로 인터페이스 및 검색 기능을 제공한다.

이를 통해 기존 방식보다 적은 비용으로 시스템의 안정성과 일관성을 높이는 데 기여할 수 있을 것으로 기대된다. 향후에는 추가적인 기능으로 확장 가능한 인프라와 로그 및 이벤트 데이터에 대한 접근 제어 및 보안 설정을 연구해 볼 계획이다.

ACKNOWLEDGMENT

"본 연구는 과학기술정보통신부 및 정보통신기획평가원의 대학 ICT 연구센터사업의 연구결과로 수행되었음" (IITP-2024-RS-2023-00258649)

참 고 문 헌

[1] Horalek, J. (2023). Proposed solution for log collection and analysis in Kubernetes environment.

[2] https://www.fluentd.org/

[3] https://www.elastic.co/kr/elasticsearch

[4] https://www.elastic.co/kr/kibana

[5] Nair, Rahul, et al. (2022). The Kubernetes Operator Framework Book. Packt.

[6] https://sdk.operatorframework.io/

[7] Abdollahi Vayghan, L. A Kubernetes controller for managing the availability of elastic microservice based stateful applications.