쿠버네티스 기반 다중 클러스터에서의 정책 기반 스케줄링 설계

김영선, 김영한*
숭실대학교

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

A Design of Policy based Scheduling in Kubernetes Multi-Cluster

Kim Young Sun, Kim Young Han*
Soongsil Univ.

요 약

쿠버네티스 클러스터는 지리적 제약, 사업자 종속성과 같은 문제점 해결 및 운영 환경 분리, 하이브리드 클라우드 구축과 같은 다양한 이유로 인해 그 수와 규모가 점점 커지고 있다. 기존 다중 클러스터에서의 쿠버네티스 리소스 배포 방법은 각 클러스터마다 일일이 배포하거나 일괄적으로 균등하게 배포한다. 본 논문에서는 정책을 이용하여 사용자의 요구에 맞게 애플리케이션이 배포될 수 있는 쿠버네티스 기반 다중 클러스터 환경에서의 정책이 적용된 스케줄링을 제안한다.

Ⅰ. 서 론

쿠버네티스는 컨테이너 기반 클라우드 인프라 오케스트레이터로 컨테이너화된 인스턴스를 관리하는 기능을 제공한다[1]. 쿠버네티스 클러스터는 마스터 노드와 여러 대의 워커 노드로 이루어져 있으며, 단일 클러스터에서 다중 클러스터로 운영이 확대되고 있다. 이를 통해 확장성 문제, 지리적 제약, 사업자 종속성과 같은 문제점을 해결할 수 있으며 하이브리드 클라우드 구축과 클러스터를 운영 환경별로 나누는 것이 가능해졌다. 그러나 쿠버네티스 다중 클러스터에 애플리케이션을 배포하고자 할 때, 일일이 수동으로 배포하거나 배포하고자 하는 클러스터에 일괄적으로 균등하게 배포하여 편의성이 떨어진다. 각 클러스터에 가중치를 주어 배포하는 방법도 있지만, 사용자가 직접 클러스터마다 설정해야 한다는 불편함이 있다.

본 논문에서는 쿠버네티스 멀티 클러스터를 위한 오픈소스 프로젝트인 KubeFed 와 정책 컨트롤러 오픈소스 프로젝트인 OPA 에 대해 소개하고 이를 적용하여 쿠버네티스 기반 다중 클러스터에서의 정책 기반 스케줄링 구조를 제안한다.

Ⅱ. KubeFed

KubeFed(Kubernetes Cluster Federation)는 여러 개의 쿠버네티스 클러스터들을 하나의 논리 클러스터인 것처럼 사용하고 여러 클러스터에 걸쳐 쿠버네티스 리소스를 배포하고 관리하는 애플리케이션 관리 기능을 제공한다[2]. 그림 1 은 KubeFed 를 사용하여 여러 개의 클러스터를 연합한 구조로 KubeFed 컨트롤 플레인이 실행되는 호스트 클러스터와 여러 개의 멤버 클러스터로 구성된다. 기존의 다중 클러스터와의 차이점은 애플리케이션 배포와 관리를 제공하는 컨트롤 플레인이 존재하여 이를 통해 연결되어 있어 워크로드 관리를 지원한다.

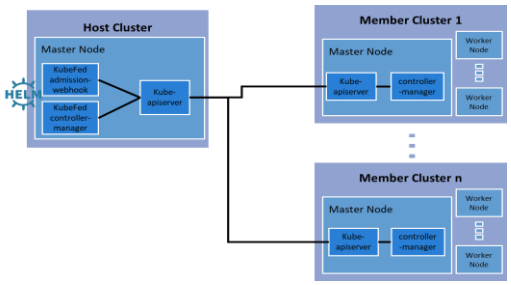

그림 1. KubeFed 구조

KubeFed 컨트롤 플레인은 CRD(Custom Resource Definition)와 컨트롤러 구조를 채택하며, 구성 요소는 표 1 과 같다. 별도의 API 서버를 구축하지 않고 기존의 쿠버네티스 kube-apiserver 를 이용하므로 이를 위한 admission controller 와 reconcile loop 를 수행하는 컨트롤러 매니저로 이루어져 있다.

표 1. KubeFed 컨트롤 플레인 구성 요소

이름	역할
KubeFed admission-webhook	CRD 를 이용하여 KubeFed 리소스를 구현하므로 kube-apiserver 의 오브젝트 생성 요청을 가로채어 제어한다.
KubeFed controller-manager	KubeFed 리소스 배포 및 관리하고 kubefed 컨피그 관리한다. 이를 위해 멤버 클러스터에서 실행되는 KubeFed 리소스와 이를 감시하고 필요한 조정을 수행하는 여러 컨트롤러를 실행한다. (클러스터 컨트롤러, 스케줄링 컨트롤러, 컨피그 컨트롤러)

Ⅲ. OPA

OPA(Open Policy Agent)는 정책을 통합하여 적용 및 관리하도록 도와주는 오픈소스 프로젝트로 서비스와 함께 배치될 수 있는 정책 엔진이며 정책은 소프트웨어 서비스의 동작을 제어하는 규칙 집합을 의미한다[3]. 이를 통해 서비스로부터 정책을 분리하여 서비스 자체와 별도로 정책을 배포하고 관리할 수 있다. OPA 는 정책 시행으로부터 정책 결정을 분리하여 그림 3 과 같은 동작 절차를 수행한다. 서비스는 OPA 에게 JSON 과 같은 구조화된 데이터를 입력으로 제공하며 정책 결정을 요청한다. OPA 는 정책과 입력받은 데이터를 평가하여 정책 결정을 생성하고 이를 다시 서비스로 전송한다. 이때, 정책 결정은 단순한 예, 아니요 또는 허용, 거부와 같은 답변에만 국한되지 않고 구조화된 데이터를 출력으로 생성할 수 있다.

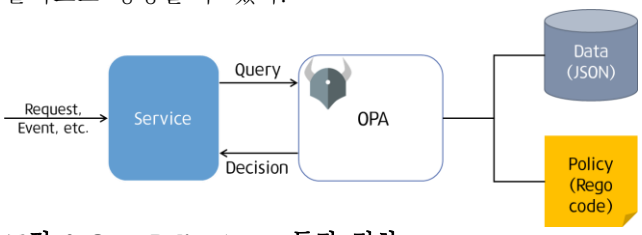

그림 2. Open Policy Agent 동작 절차

Ⅳ. 다중 클러스터에서의 정책 기반 스케줄러 설계

본 논문에서는 KubeFed 를 이용하여 연합된 쿠버네티스 다중 클러스터 환경에서의 스케줄링 작업에 OPA 를 적용하여 정책 기반 스케줄링을 제안한다. 그림 3 은 쿠버네티스 API 서버 동작 절차에 KubeFed 리소스를 위한 admission controller 와 OPA 를 admission controller 로 사용하여 추가한 절차이다. 쿠버네티스 API 서버 동작 절차에서의 mutating admission 은 들어온 요청에 대해 변경하기 위해 사용하고 validating admission 은 들어온 요청에 대해 유효성을 검증하기 위해 사용한다. KubeFed 리소스는 CRD 를 이용하여 구현되기 때문에 mutating admission 컨트롤러가 필요하며, OPA 는 정책 컨트롤러로 mutating admission 과 validating admission 에 추가한다.

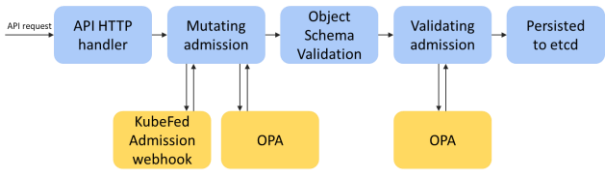

그림 3. admission controller 를 적용한 쿠버네티스 API 서버 동작 절차

그림 4 는 KubeFed 리소스를 배포할 때의 동작 절차로 어떤 정책을 적용하는지에 따라 스케줄링 결과가 달라진다. 그림 4 의 경우에는 배포하는 애플리케이션에 라벨링을 하고 클러스터는 특성별로 묶어 라벨 값에 따라 어떤 클러스터에 배포되는지를 결정된다. KubeFed 리소스 생성 요청이 들어오면 kube-apiserver 는 HTTP 핸들러를 생성하고 KubeFed Admission webhook 이 생성 요청을 가로채어 제어한다. 다음으로 첫 번째 OPA 가 애플리케이션의 라벨을 값을 확인하여 정책에 맞게 애플리케이션의 클러스터 목록을 수정한다. 해당 오브젝트의 스키마가 유효한지 검사한 후 마지막으로 두 번째 OPA 를 이용하여 배포한 정책에 적합한지 검사한 후 결과에 따라 리소스를 생성하거나 생성을 거부한다. 마지막 유효성 검사의 경우에는 신뢰할 수 있는 지정된 이미지 레지스트리를 사용 여부를 확인하거나 위반 행위 여부를 확인하여 리소스 생성 여부를 결정할 수 있다. 그림 4 의 경우 이외에도 애플리케이션에 데이터의 중요성에 따라 라벨링하고 그 값에 따라 퍼블릭 클라우드와 프라이빗 클라우드를 나누어 배포할 수 있다.

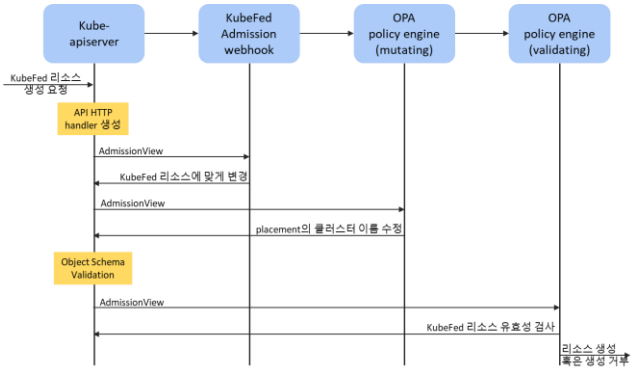

그림 4. KubeFed 리소스 배포 동작 절차

Ⅳ. 결론

본 논문에서는 KubeFed 를 이용하여 쿠버네티스 다중 클러스터를 연합하고 다중 클러스터 환경에서의 쿠버네티스 리소스 스케줄링 작업에 OPA 를 적용한 정책 기반 스케줄링을 제안하였다. 다중 클러스터 환경에 쿠버네티스 리소스를 배포할 때, 일괄 배포 및 각 클러스터에 가중치를 지정하여 배포하는 것이 가능하지만, 배포하고자 하는 애플리케이션이 다양하고 클러스터의 분류가 다양할수록 자동화는 불가피해진다. 본 논문에서 제안한 정책 기반 스케줄링은 스케줄링 작업에 정책이 분리되어 서비스와 별도로 배포 및 관리가 가능해지고 후에 변화하는 요구사항에 맞게 조정하는 것이 가능할 것으로 예상된다. 추후 연구에서 실제 기능들을 설계하고 구현하고자 한다.

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, https://kubernetes.io/, 2021

[2] KubeFed, https://github.com/kubernetes-sigs/kubefed, 2021

[3] Open Policy Agent, https://www.openpolicyagent.org, 2021