한국컴퓨터정보학회 동계학술대회 논문집 제32권 제1호 (2024. 1)

도커와 쿠버네티스 기반 미세먼지 데이터 수집 방안

최효현*, 김연욱O
*인하공업전문대학 컴퓨터정보공학과,
O인하공업전문대학 컴퓨터정보공학과
e-mail: hchoi@inhatc.ac.kr*, ccoma511@naver.comO

Docker and Kubernetes Based Approaches for PM Data Collection

Hyo Hyun Choi*, Yeon Wook KimO
*Dept. of Computer Science & Engineering, Inha Technical College,
ODept. of Computer Science & Engineering, Inha Technical College

● 요 약 ●

본 논문에서는 도커와 쿠버네티스를 활용하여 미세먼지 데이터를 수집할 때 다량으로 늘어나는 데이터를 효율적으로 수집하고 관리하기 위한 방안을 제시한다. 도커 이미지는 작성된 Dockerfile을 통해 생성되며, 필요한 의존성과 설정이 반영되어 있다. 쿠버네티스를 이용하여 생성된 도커 이미지를 기반으로 컨테이너를 생성하고, 컨테이너들을 파드 내에서 실행함으로써 데이터를 효율적으로 수집하고 관리한다.

키워드: 키워드: 미세먼지(particulate matter), 크롤링(Crawling), 공공데이터 API(Public Data API), 도커(Docker), 쿠버네티스(Kubernetes)

I. Introduction

현재 공기 중의 미세먼지 농도가 높아짐에 따라 환경과 건강에 미치는 영향이 커지고 있다. 높은 미세먼지 농도는 호흡기 질환의 증가와 기후 변화에 대한 부정적인 영향을 초래할 수 있다. 이러한 미세먼지 농도를 효과적으로 낮추기 위한 기술적인 해결책을 탐구하고자 한다. 이를 위해서는 정확하고 실시간인 미세먼지 데이터 수집이 필수적이다.

본 논문에서는 Selenium을 이용한 크롤링과 공공데이터 API를 활용하여 미세먼지 데이터를 수집한다. 이때 크롤링과 API 호출은 많은 시간이 소요되며, 데이터 양이 많아질수록 효율적으로 저장하고 관리하기 어려운 문제가 있다. 이를 해결하기 위해 Docker를 활용한 컨테이너화와 Kubernetes를 이용한 오케스트레이션을 통해 효율적으로 데이터를 수집하고 관리하며, 확장 가능한 환경을 구성하고자 한다.

II. Preliminaries

1. Selenium

Selenium은 웹 응용 프로그램을 크롤링하고 자동으로 제어하는데 사용되는 도구이다. 다양한 프로그래밍 언어를 지원하며, 본 논문에서는 Java를 기반으로 한 Selenium을 활용하여 동적인 웹 페이지에서 미세먼지 측정소에 대한 데이터를 수집한다.

2. Public Data API

Public Data API는 정부에서 제공하는 프로그래밍 인터페이스로, 공공데이터 포털의 오픈 API이다. 이를 통해 한국환경공단인 에어코리아로부터 대기오염정보를 수집할 수 있다. 본 논문에서는 Java를 기반으로 API를 호출하며 측정소별 실시간 미세먼지 데이터를 수집한다.

3. Docker

Docker는 컨테이너 기술을 활용하여 각각의 구성 요소를 격리된 환경에서 실행함으로써 시스템의 유연성을 향상시킨다. 이를 통해 개발 환경과 운영 환경 간의 일관성을 유지할 수 있으며 데이터 수집 시스템을 효율적으로 운영할 수 있다.

4. Kubernetes

Kubernetes는 Docker Container를 효율적으로 관리한다. 이는 컨테이너의 배포, 확장, 자동 복구 등을 자동화함으로써 시스템의 안정성과 확장성을 향상시킬 수 있다.

III. Development

먼저 Java 기반의 Selenium 라이브러리를 추가하고 웹 브라우저는 Chrome을 사용하기에 버전에 맞는 ChromeDriver를 사용하여 측정소에 대한 데이터를 크롤링 해온다.

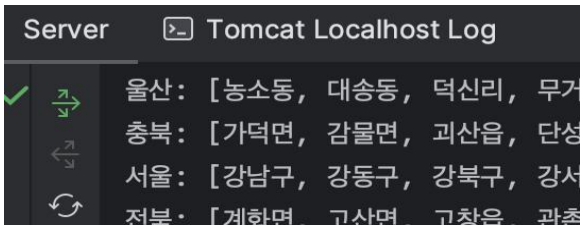

Fig. 1. Crawling Result

Fig. 1은 미세먼지 측정소에 대한 데이터를 웹 크롤링 결과의 일부이다. 크롤링하여 얻은 미세먼지 측정소 데이터를 공공데이터 API 호출 시에 파라미터로 사용하여 각 측정소별 실시간 미세먼지 측정 데이터를 수집해온다.

다량의 데이터를 효율적으로 저장하고 관리하기 위해 측정소별 지역을 기준 잡아 미세먼지 데이터를 저장하고 관리하고자 한다. Docker Image를 생성하기 위해 Dockerfile을 작성 후 Build 하여 이미지를 생성한다.

Fig. 2. Docker Images

Fig. 2는 측정소별 데이터를 저장하는 기능과 컨테이너를 실행하기 위한 파일 시스템 설정이 들어가 있는 이미지 파일이다. 임시로 측정소별 지역을 3개만 선택하여 이에 맞는 Docker Image를 생성하였다.

위에서 생성한 Docker Image를 Kubernets를 사용하여 컨테이너를 생성하고 관리한다. 데이터의 영속성을 보장하고 데이터의 백업 및 복구를 위해 Volume을 사용한다.

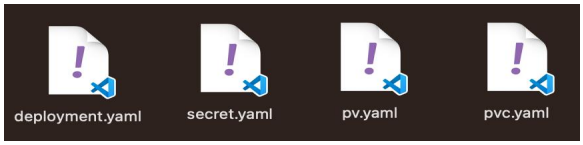

Fig. 3. Kubernetes YAML File

Fig. 3은 본 논문에서 사용된 네 가지의 YAML 파일이다. deployment.yaml 파일은 파드의 수, 컨테이너 이미지, 환경 변수 등이 작성되어 있는 텍스트 파일이다. secret.yaml 파일은 데이터베이스의 비밀번호와 사용할 데이터베이스 명을 Base64로 암호화하여 작성되어 있다. pv.yaml 파일은 Persistent Volume으로 클러스터 내에서 사용할 수 있는 영구적인 스토리지를 정의하는 텍스트 파일이다. 로컬의 위치를 마운트 하여 Volume으로 사용할 수 있도록 작성하였다. pvc.yaml 파일은 Persistent Volume Claim으로 클러스터에서 PV를 사용하는 애플리케이션에 스토리지를 동적으로 할당하는 데 사용되는 텍스트 파일이다.

IV. Conclusions

필요한 미세먼지 데이터를 가져오기 위해서 많은 시간이 소요되는 문제를 해결하기 위해 다량의 미세먼지 데이터를 저장하기로 하였다. 지속적으로 늘어나는 다량의 데이터를 효율적으로 관리하기 위해 도커와 쿠버네티스를 사용하였으며, 이는 시스템의 환경을 일관성 있게 유지할 수 있으며 컨테이너의 배포, 확장, 자동 복구 등을 자동화하여 시스템의 안정성과 확장성을 향상시켜 주었다. 향후에는 수집된 미세먼지 데이터를 활용해 미세먼지 감축 시스템을 구현할 계획이다.

REFERENCES

[1] Crawling Web Site, https://www.airkorea.or.kr/web/stationInfo?pMENU_NO=93
[2] Public Data API (PM), https://www.data.go.kr/data/15073861/openapi.do
[3] Docker's official website, https://www.docker.com
[4] Kubernetes's official website, https://kubernetes.io