

# 클라우드 환경에서 오토 스케일링이 가능한 센서 데이터 수집 시스템 설계

박수용\*, 최수민\*, 신용태\*\*  
 \*승실대학교 컴퓨터학과  
 \*\*승실대학교 컴퓨터공학부  
 tndyd5390@naver.com

## A Study on Tools for Agent System Development

Soo-Yong Park\*, Su-Min Choi\*, Yong-Tae Shin\*\*  
 \*Dept of Computer Science, Soongsil University

### 요 약

센서 네트워크의 센서 개수가 늘어나고 데이터 수집 주기가 짧아지며 데이터의 용량도 늘어남에 따라 데이터를 수집하는 중앙서버의 과부하가 걸리는 현상이 발생할 수 있다. 본 논문에서 제안하는 시스템은 센서 데이터를 수집하는 모듈을 컨테이너화 하여 쿠버네티스로 관리한다. 또한 쿠버네티스의 오토 스케일링 기능을 이용하여 데이터 수집 모듈의 과부하가 발생할 경우 자동으로 수집 모듈을 복사하여 scale out 할 수 있다.

### 1. 서론

센서 네트워크는 특정 지역에 분산하여 배치된 센서로부터 데이터를 수집하는 네트워크를 의미한다. 센서 네트워크는 환경 모니터링, 보안 및 감시, 스마트 홈, 스마트 그리드 등의 다양한 분야에서 사용되고 있다. 센서 네트워크는 센서가 수집한 데이터를 게이트웨이를 통하여 중앙의 서버로 전송하는 방식을 사용하고 있다. 센서 네트워크를 사용하는 분야가 다양해지고 측정하고자 하는 지역이 넓어짐에 따라 센서의 수가 늘어났다. 따라서 중앙 서버로 전송되는 트래픽과 데이터의 양이 증가하고 중앙 서버에 과부하가 걸리는 현상이 발생하게 된다[1].

제안하는 시스템은 클라우드 환경에서 다수의 서버를 이용하여 클러스터링을 수행한다. 각각의 서버에는 도커를 이용하여 데이터 수집 시스템을 구축하고 센서로부터 전송되는 트래픽과 데이터의 양을 지속적으로 모니터링한다. 센서 네트워크에서의 어떠한 이벤트로 인해 트래픽 또는 데이터의 양이 증가하여 도커의 자원이 부족해지면 클러스터에서 동일한 도커를 하나 더 생성하여 부하를 분산시킬 수 있다.

본 논문의 구성은 다음과 같다. 2장 관련 연구에서는 시스템 설계에 필요한 기반 기술인 도커와 쿠

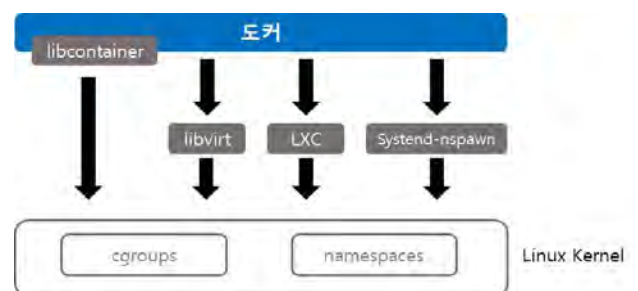
버네티스에 대해 연구한다. 3장 시스템 설계에서는 관련 연구를 기반으로 시스템을 설계한다. 4장 결론에서는 설계된 시스템에 기초하여 결론을 제시한다.

### 2. 관련 연구

본 장에서는 클라우드 환경에서 오토스케일링이 가능한 센서 데이터 수집 시스템 설계에 필요한 기반 기술인 도커와 쿠버네티스에 대해 연구한다.

#### 2.1. 도커

도커는 클라우드 환경에서 애플리케이션을 처리할 수 있는 컨테이너 플랫폼을 의미한다. 도커는 운영체제에서 namespace와 cgroup 기능을 이용하여 프로세스에서 사용하는 자원을 분리하여 컨테이너라는 공간에 할당한다[2]. (그림 1)은 도커에서 사용하는 자원 분리 방식을 나타낸다.

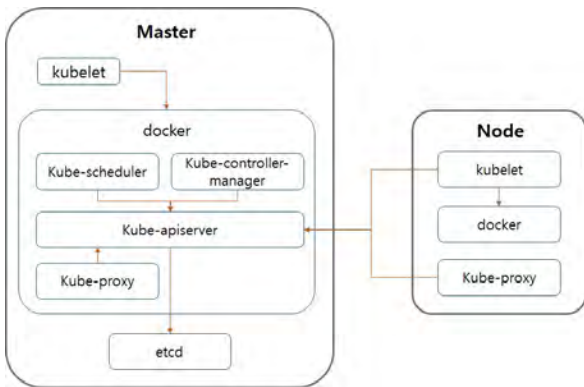


(그림 1) 도커에서 사용하는 자원 분리 방식

컨테이너에서는 할당된 자원을 활용하여 애플리케이션을 실행할 수 있고 애플리케이션의 실행 상태를 이미지로 저장하여 사용자가 원할 때 바로 실행이 가능하다. 이미지로 저장하고 원할 때 실행한다는 점에서 다른 오픈소스 가상화 소프트웨어와 비슷하지만 오픈소스 가상화 소프트웨어는 운영체제를 가상화하고 도커는 자원을 격리시킨다는 점에서 분명한 차이가 있다. 오픈소스 가상화 소프트웨어는 가상화된 머신에 항상 게스트 OS를 설치해야 하기 때문에 용량이 많이 필요하다. 도커는 실행파일을 호스트에서 직접 실행하기 때문에 가상화에 비해 저장공간이 적게 필요하다.

### 2.2. 쿠버네티스

쿠버네티스는 구글에서 개발하고 배포한 컨테이너 작업을 자동화하는 컨테이너 오케스트레이션 오픈소스 플랫폼이다. 도커를 이용하여 컨테이너를 작성하고 실행하는데 필요한 수동 프로세스를 자동으로 전환할 수 있다. (그림 2)는 쿠버네티스의 아키텍처를 나타낸다.



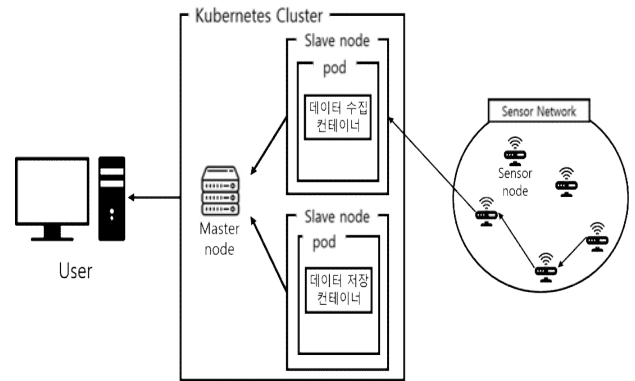
(그림 2) 쿠버네티스의 아키텍처

쿠버네티스는 마스터 노드와 슬레이브 노드로 구성된다[2]. 마스터 노드는 컨테이너를 생성하고 슬레이브 노드에 배포하며 관리하는 역할을 한다. 슬레이브 노드는 마스터 노드로부터 명령을 전달받고 실제 실행해야 할 애플리케이션이 담긴 컨테이너를 실행하는 노드이다. 또한 슬레이브 노드는 현재 실행 상태인 컨테이너의 정보를 마스터 노드로 전송하는 역할을 한다. 마스터 노드는 슬레이브 노드의 자원 사용량을 정할 수 있고 자원 사용량을 넘어서는 컨테이너를 자동으로 복사해서 다른 노드에 실행시키

고 로드밸런싱을 수행하는 오토스케일링 기능을 가지고 있다.

### 3. 시스템 설계

제안하는 시스템은 센서로부터 생성되는 트래픽의 양과 데이터에 따라 수집 서버를 scale out 할 수 있어야 한다. (그림 3)은 제안하는 시스템의 아키텍처를 나타낸다.



(그림 3) 제안하는 시스템의 아키텍처

3개 이상의 서버를 이용하여 쿠버네티스 클러스터를 구축한다. 3개 이상의 서버를 사용하는 이유는 마스터 서버가 과부하 되어 제 기능을 수행하지 못할 경우 다른 마스터 서버를 선출해야 하기 때문이다. 클러스터 구축이 완료되면 센서 네트워크의 데이터 수집 모듈과 저장 모듈을 도커를 사용하여 컨테이너로 생성한다. 생성된 컨테이너는 마스터 노드에 의해 슬레이브 노드로 배포되고 데이터 수집 및 저장을 시작한다. 슬레이브 노드는 데이터를 수집하고 있는 컨테이너의 상태를 지속적으로 마스터에게 전송한다. 마스터 노드는 데이터 수집 컨테이너의 자원 사용량이 일정 수준을 넘어서면 데이터 수집 컨테이너를 복사하고 다른 슬레이브 노드에 배포한다. 이때 마스터 노드에서는 동일한 작업을 수행하는 컨테이너를 여러 개 생성하기 때문에 로드밸런싱이 필요하다[3].

쿠버네티스에서 오토스케일링 및 로드밸런싱 기능을 사용하기 위해서는 metrics-server를 설치해야 한다. metrics-server는 컨테이너의 자원 사용량을 지속적으로 모니터링하는 서버로 전체 클러스터의 특성을 검사하여 클러스터 내의 애플리케이션의 성능을 검사할 수 있다. metrics-server 서버 설치 후에 컨테이너의 메트릭 데이터를 수집하기 위하여

metrics-server-deployment.yaml 파일을 수정해야 한다. (그림 4)는 수정된 설정 파일을 나타낸다.

```

1 containers:
2 #수정하고자 하는 인스턴스의 이름
3 - name: metrics-server
4   image: k8s.gcr.io/metrics-server-amd64:v0
5   args:
6     #데이터 수집을 위한 인증서가 저장된 폴더
7     --cert-dir/tmp
8     #메트릭 데이터 수집시 사용하는 포트
9     --secure-port=4443
10    #인증서로 인한 오류방지
11    --kubelet-insecure-tls
12    #마스터 노드가 InternalIP를 사용하여 다른
13    #노드를 검색할 수 있도록 함
14    --kubelet-preferred-address-types=Internal

```

(그림 4) 수정된 metrics-server의 설정파일

metrics-server는 메트릭 데이터의 수집만을 담당하고 실제로 컨테이너는 scale out하지 않는다. 실제 컨테이너의 scale out을 위해서는 autoscaling.yaml 파일을 생성하여 적용시켜야 한다. (그림 5)는 autoscaling.yaml 파일을 나타낸다.

```

1 #생성된 인스턴스의 종류 여기서는 autoscaler
2 kind: HorizontalPodAutoscaler
3 metadata:
4   #autoscaler의 이름
5   name: autoscaler
6 #autoscaler의 세부 설정
7 spec:
8   #컨테이너의 최대 복사본의 개수
9   maxReplicas: 10
10  #컨테이너의 최소 복사본의 개수
11  minReplicas: 1
12  scaleTargetRef:
13    kind: Deployment
14    name: <데이터 수집 컨테이너의 이름>
15  #cpu 사용률이 30%가 넘으면 스케일링 함
16  targetCPUUtilizationPercentage: 30

```

(그림 5) autoscaling.yaml 파일의 내용

autoscaling.yaml 파일을 적용시키면 마스터 노드는 데이터 수집 컨테이너의 cpu 사용률이 30%가 넘을 경우 복사본을 생성하여 다른 슬레이브 노드로

scale out 할 수 있다.

#### 4. 결론

본 논문에서는 도커와 쿠버네티스를 이용하여 센서 네트워크에서 수집하는 데이터의 양이 많아질 경우 자동으로 수집 모듈을 scale out 하여 더 안정적으로 데이터를 수집할 수 있는 시스템을 설계하였다. 데이터를 저장하기 위한 저장 모듈 또한 같은 방식으로 scale out 가능하다. 그러나 데이터의 저장소가 scale out 되어 복사가 될 경우 같은 데이터가 중복되기 때문에 저장소를 제외한 모듈만 복사하는 연구가 추가적으로 이루어져야 한다. 또한 본 논문에서는 시스템의 cpu 사용률이 30%가 넘을 경우 오토 스케일링을 실행하도록 설정하였으나 추후 실험을 통하여 가장 효율적인 오토스케일링 시점을 찾아야한다.

#### Acknowledgement

본 논문은 2019년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (No.IITP-2019-0-00135 ,ICT 기반 환경 모니터링 센서 신뢰성 검증 및 평가 플랫폼)

#### 참고문헌

[1] 장시웅, 김지성 "대규모 센서 네트워크에서 센서 데이터 수집을 위한 효율적인 통신 시스템 설계 및 구현", 한국정보통신학회논문지, 제24권, 제1호, pp113-119, 2020

[2] 장현준, 임인구, 진현욱, "도커 컨테이너 자원 활용률 모니터", 한국정보과학회 학술발표논문집, 2019, pp1086-1088

[3] 김경일, "클라우드 서비스를 위한 쿠버네티스 구조", 한국통신학회지, 제35권, 제11호, pp11-19, 2018

[4] Ilseok Han, Jonggyu Park, Wonyoul Bae, Hagbae Kim, "Development of an Optimal Load Balancing Algorithm based on ANFIS modeling in a Clustering Web Server", 한국통신학회 학술대회 논문집, 2003, pp1215-1218