



빅데이터 모니터링 시스템 구축

오늘의 학습내용

- 쿠버네티스 로깅 아키텍처 구축
- 컨테이너 기반 로그관제 시스템 구축

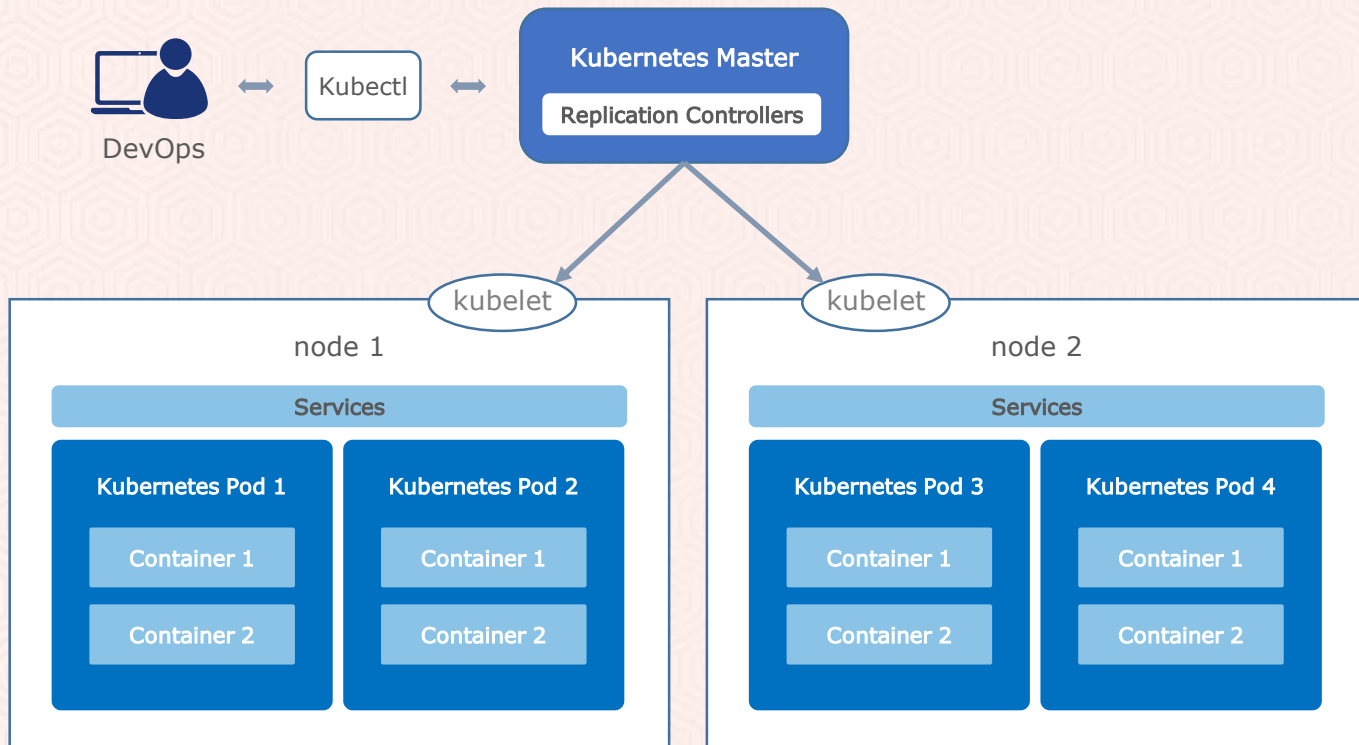
◆ 쿠버네티스 로깅 아키텍처 구축

■ 쿠버네티스 개요

- ❖ Horizontal scaling : 컨테이너는 가상화가 아닌 격리를 사용하여 부팅 과정이 없고 작은 이미지를 사용하여 복제, 가용성이 필요할 때 컨테이너를 쉽게 확장 가능
- ❖ Service discovery and load balancing : 마이크로서비스 아키텍처나 일반 애플리케이션에서도 각 서비스나 애플리케이션 확장 시 골고루 부하를 분산하는 로드밸런싱과 각 애플리케이션을 찾아주는 서비스 디스커버리는 필수 요소, 쿠버네티스는 ReplicaSet 을 통해 운영 플랫폼 자체에서 서비스 디스커버리와 로드밸런싱을 기본 제공
- ❖ Rollout and rollback : 무중단 배포와 배포 롤백 및 특정 버전으로 돌아가는 기능은 서비스 연속성과 장애 최소화를 위해 필수적인 요구사항, 쿠버네티스는 배포에서 ReplicaSet 을 관리해 정책에 따른 무중단 배포, 업데이트 등을 플랫폼 차원에서 지원
- ❖ Self-healing : 컨테이너의 메인 프로세스가 종료되어 정지된 컨테이너를 다시 시작하고, 기 정의된 API 등으로 컨테이너 상태를 파악해 정상 기능을 하지 못하는 컨테이너를 재 시작 하는 작업도 쿠버네티스가 관리
- ❖ Secret and configuration management : 쿠버네티스는 데이터베이스 접속정보나 타 시스템 인증 정보 등 안전하게 관리해야 하는 비밀 정보를 안전하게 보관하고 사용하는 secret 오브젝트 및 운영에 필요한 환경 변수를 저장하는 configmap 오브젝트를 지원

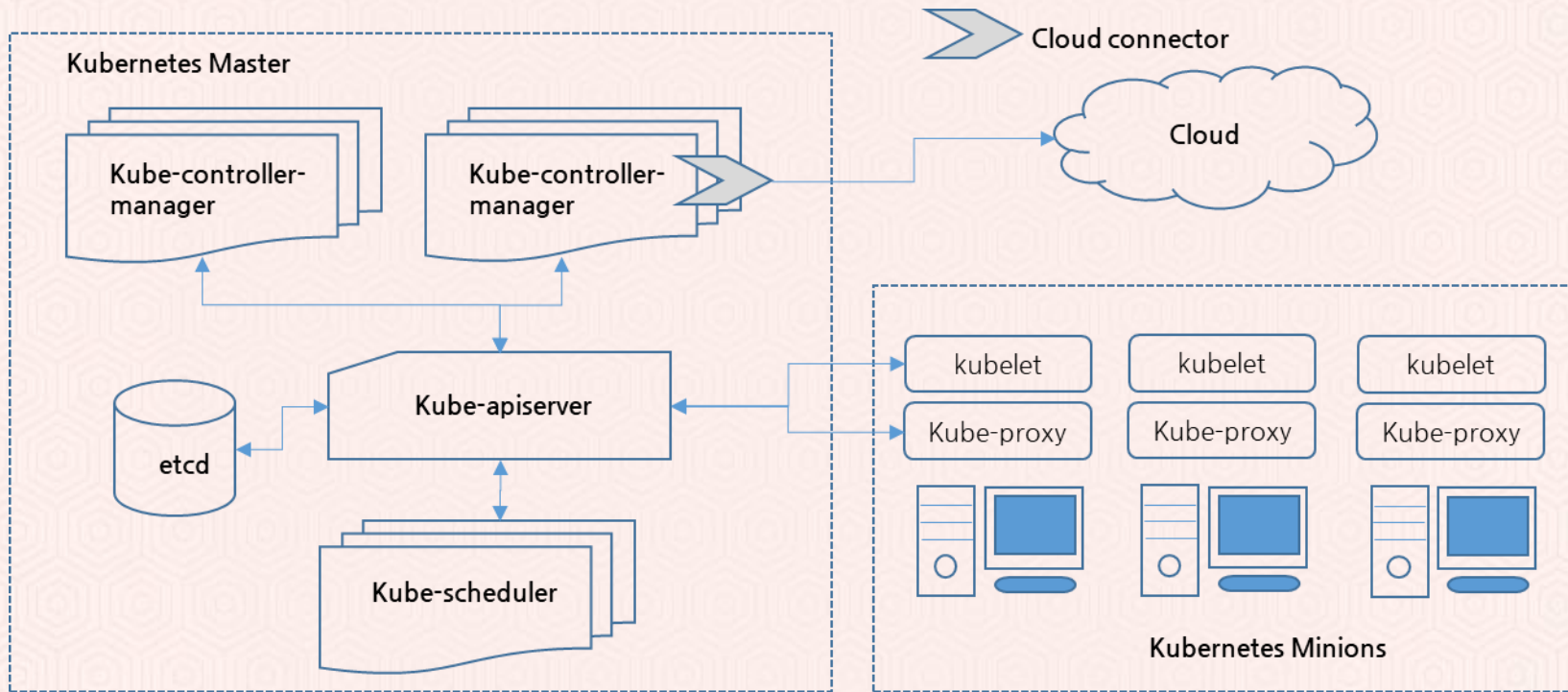
◆ 쿠버네티스 로깅 아키텍처 구축

■ 쿠버네티스와 컨테이너 구조



◆ 쿠버네티스 로깅 아키텍처 구축

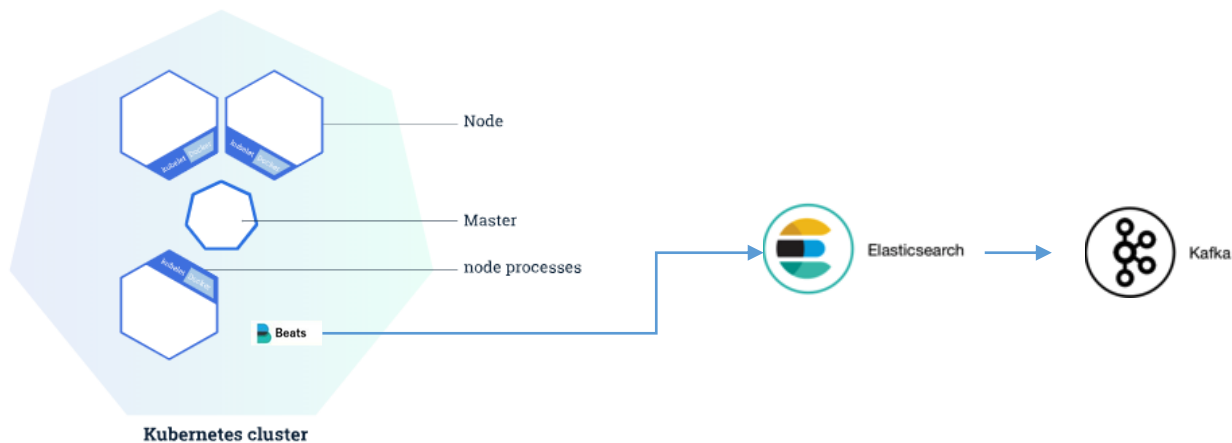
■ 쿠버네티스 아키텍처



◆ 쿠버네티스 로깅 아키텍처 구축

■ 쿠버네티스 로깅 시스템 구성

- ❖ 수집된 Log 데이터를 저장 및 조회하는 Elasticsearch
- ❖ 쿠버네티스의 각 node에서 daemonset 기반으로 log를 수집하여 elasticsearch로 전송하는 Filebeat
- ❖ 수집한 log를 visualize하는 Kibana



▲ 쿠버네티스 로깅 아키텍처 구축

■ 쿠버네티스 로깅 시스템 구성

❖ filebeat-kubernetes.yaml

```
- condition.contains:
  kubernetes.labels.app: redis
config:
  - module: redis
    log:
      input:
        type: docker
        containers.ids:
          - ${data.kubernetes.container.id}
      slowlog:
        enabled: true
        var.hosts: ["${data.host}:${data.port}"]
```

쿠버네티스 로깅 아키텍처 구축

쿠버네티스 로깅 시스템 구성

❖ Kubernetes 로그, 메트릭, APM

Kubernetes 로그Kubernetes 메트릭Kubernetes APMOpenTracingJaeger 추적

Logs

kubernetes.pod.uid: f2046b20-a4d8-11e9-8210-42010a8e0164

Timestamp	event.dataset	Message
Jul 12, 2019 @ 15:58:49.491	redis.log	[redis.log][warning]
Jul 12, 2019 @ 15:58:49.492	redis.log	[redis.log][notice]
Jul 12, 2019 @ 15:58:49.493	redis.log	[redis.log][notice]
Jul 12, 2019 @ 15:51:50.647	redis.log	[redis.log][warning]
Jul 12, 2019 @ 15:51:50.648	redis.log	[redis.log][notice]
Jul 12, 2019 @ 15:51:50.649	redis.log	[redis.log][notice]
Jul 12, 2019 @ 15:52:51.799	redis.log	[redis.log][warning]
Jul 12, 2019 @ 15:52:51.799	redis.log	[redis.log][notice]
Jul 12, 2019 @ 15:52:51.800	redis.log	[redis.log][notice]
Jul 12, 2019 @ 15:53:52.957	redis.log	[redis.log][warning]
Jul 12, 2019 @ 15:53:52.957	redis.log	[redis.log][notice]
Jul 12, 2019 @ 15:53:52.958	redis.log	[redis.log][notice]
Jul 12, 2019 @ 15:54:53.108	redis.log	[redis.log][warning]
Jul 12, 2019 @ 15:54:53.109	redis.log	[redis.log][notice]
Jul 12, 2019 @ 15:54:53.110	redis.log	[redis.log][notice]
Jul 12, 2019 @ 15:55:54.285	redis.log	[redis.log][warning]
Jul 12, 2019 @ 15:55:54.285	redis.log	[redis.log][notice]
Jul 12, 2019 @ 15:55:54.287	redis.log	[redis.log][notice]
Jul 12, 2019 @ 15:56:55.460	redis.log	[redis.log][warning]
Jul 12, 2019 @ 15:56:55.460	redis.log	[redis.log][notice]
Jul 12, 2019 @ 15:56:55.461	redis.log	[redis.log][notice]
Jul 12, 2019 @ 15:57:56.611	redis.log	[redis.log][warning]
Jul 12, 2019 @ 15:57:56.611	redis.log	[redis.log][notice]
Jul 12, 2019 @ 15:57:56.612	redis.log	[redis.log][notice]

No additional entries found Load again

Log event document details

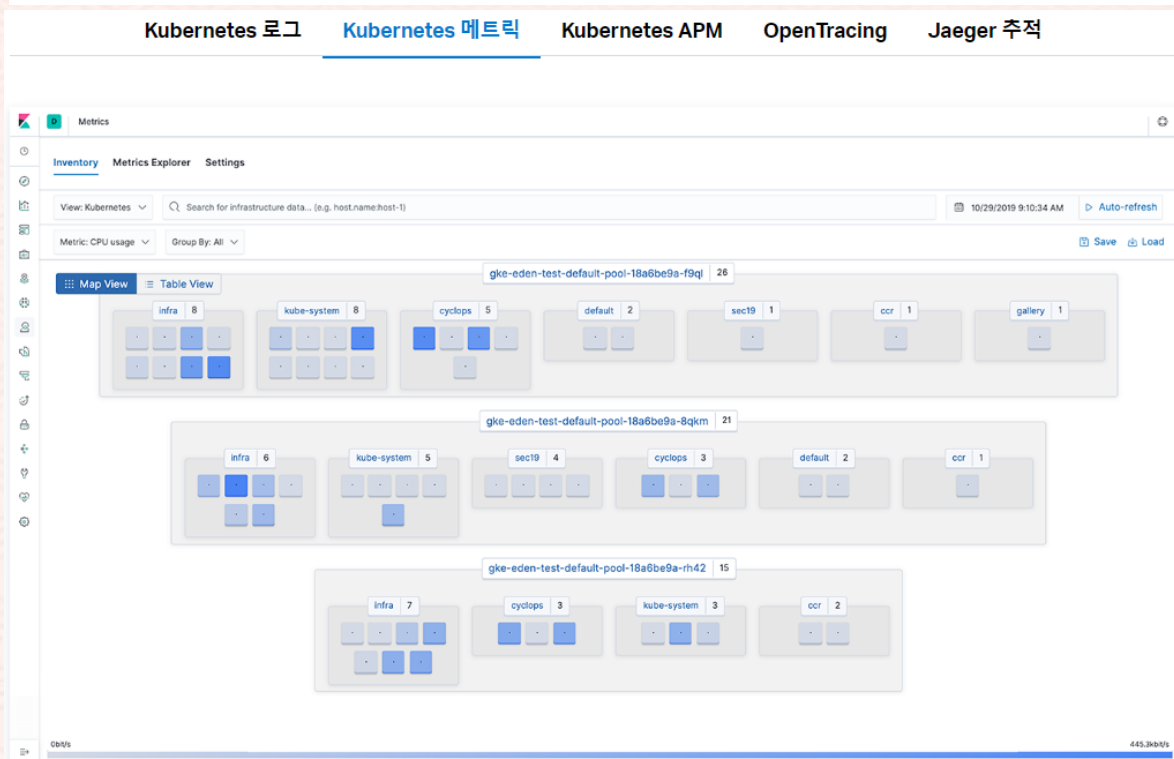
Actions

Field	Value
@timestamp	2019-07-12T19:55:54.287Z
_id	-x_BS2sBAOoy4w2krX2H
_index	filebeat-7.2.0-2019.07.10-000001
agent.ephemeral_id	5a20c2e1-f35c-4896-8cac-4dd25cfe3a6f
agent.hostname	filebeat-dynamic-xzfmn
agent.id	552da04f-2e32-440c-8ee2-51c511cd9233
agent.type	filebeat
agent.version	7.2.0
cloud.availability_zone	us-east1-b
cloud.instance.id	5328326164652724159
cloud.instance.name	gke-jamie-pmm-default-pool-625d4cbf-b8bz
cloud.machine.type	n1-standard-2

쿠버네티스 로깅 아키텍처 구축

쿠버네티스 로깅 시스템 구성

❖ Kubernetes 로그, 메트릭, APM



◆ 컨테이너 기반 로그관제 시스템 구축

■ ELK 구축

- ❖ Beats : Fabric 네트워크의 로그는 파일 형태로 저장, 로그 수집을 위해서는 Filebeat 필요
- ❖ Logstash : 다양한 플러그인을 이용하여 데이터 집계 및 보관, 서버 데이터 처리
- ❖ Elasticsearch : 대량의 데이터를 신속하고 거의 실시간으로 저장, 검색 및 분석 가능
- ❖ Kibana : 데이터를 시각화해주는 도구
- ❖ Filebeat/Metricbeat, Logstash, Elasticsearch 구성

◆ 컨테이너 기반 로그관제 시스템 구축

■ Beats 구성

❖ 설치

```
curl -L -O https://artifacts.elastic.co/downloads/beats/metricbeat/metricbeat-6.7.2-linux-x86_64.tar.gz
```

```
tar xzvf metricbeat-6.7.2-linux-x86_64.tar.gz
```

```
cd metricbeat-6.7.2-linux-x86_64
```

❖ 소유권 변경(실행 계정과 소유 계정이 동일하게 설정)

```
sudo chown -R root:root .
```

❖ Kibana Dashboard(metricbeat) 초기화(템플릿 로드)

```
sudo ./metricbeat setup -e
```

❖ 구동

```
sudo ./metricbeat -e
```

◆ 컨테이너 기반 로그관제 시스템 구축

■ filebeat 구성

❖ Docker 컨테이너의 로그를 수집하기 위해 filebeat 구성, Docker 컨테이너의 로그는 파일로 저장

#filebeat.yml 파일

```
filebeat.autodiscover:
  providers:
    - type: docker
      hints.enabled: true
      json.message_key: log
      templates:
        - condition:
            equals:
              docker.container.labels.filebeat_enable: "true"
      config:
        - type: docker
          containers.ids:
            - "${data.docker.container.id}"
          json.keys_under_root: true
          json.add_error_key: false
  processors:
    - add_cloud_metadata: ~
    - add_docker_metadata: ~
  output.logstash:
    hosts: ["${LOGSTASH_HOST}"]
```

◆ 컨테이너 기반 로그관제 시스템 구축

- filebeat을 docker로 실행하기 위해 docker-compose 파일을

❖ docker-compose-filebeat.yml

```
version: '2'

networks:
  byfn:

services:
  filebeat:
    image: docker.elastic.co/beats/filebeat:6.7.2
    container_name: filebeat
    hostname: filebeat # Docker Hostname
    environment:
      - LOGSTASH_HOST=node1:5044 # Logstash Host(extra_hosts에서 정의한 node1의 IP로 연결)
    user: root # root 계정 필요
```


◆ 컨테이너 기반 로그관제 시스템 구축

- filebeat을 docker로 실행하기 위해 docker-compose 파일을

❖ docker-compose-filebeat.yml

volumes:

- # Filebeat 설정 파일

- ./filebeat.yml:/usr/share/filebeat/filebeat.yml

- # for accessing docker

- /var/run/docker.sock:/var/run/docker.sock

- /var/lib/docker/containers:/var/lib/docker/containers:ro

extra_hosts:

- "node1:192.168.1.x" # Logstash Host IP(자신의 IP 주소에 맞게 설정. ex: 192.168.1.x)

networks:

- byfn

◆ 컨테이너 기반 로그관제 시스템 구축

■ filebeat을 구동

- ❖ `docker-compose -f docker-compose-filebeat.yml up`
- ❖ byfn 네트워크와의 연결
- ❖ `/home/hyperledger/fabric-samples/first-network/base/peer-base.yaml` 파일
-environment:
 - `FABRIC_LOGGING_FORMAT=json`
- ❖ 실행
 - `/home/hyperledger/fabric-samples/first-network/byfn.sh up`

🔺 컨테이너 기반 로그관제 시스템 구축

■ Kibana 설정

❖ Index pattern : docker_log_index-*

❖ Time Filter field name : @timestamp

Kibana

localhost:5601/app/kibana#/management/kibana/index_g=()

Help us improve the Elastic Stack by providing usage statistics for basic features. We will not share this data outside of Elastic. Read more

Yes No

Create index pattern

★ metricbeat-*

Create index pattern

Kibana uses index patterns to retrieve data from Elasticsearch indices for things like visualizations.

☐ Include system indices

Step 1 of 2: Define index pattern

Index pattern

docker_log_index-*

You can use a * as a wildcard in your index pattern.
You can't use spaces or the characters \, /, ?, *, <, >, |.

> Next step

✓ Success! Your index pattern matches 2 indices.

docker_log_index-2019.05.16

docker_log_index-2019.05.17

Rows per page: 10