Data Dashboard Workflow 사례

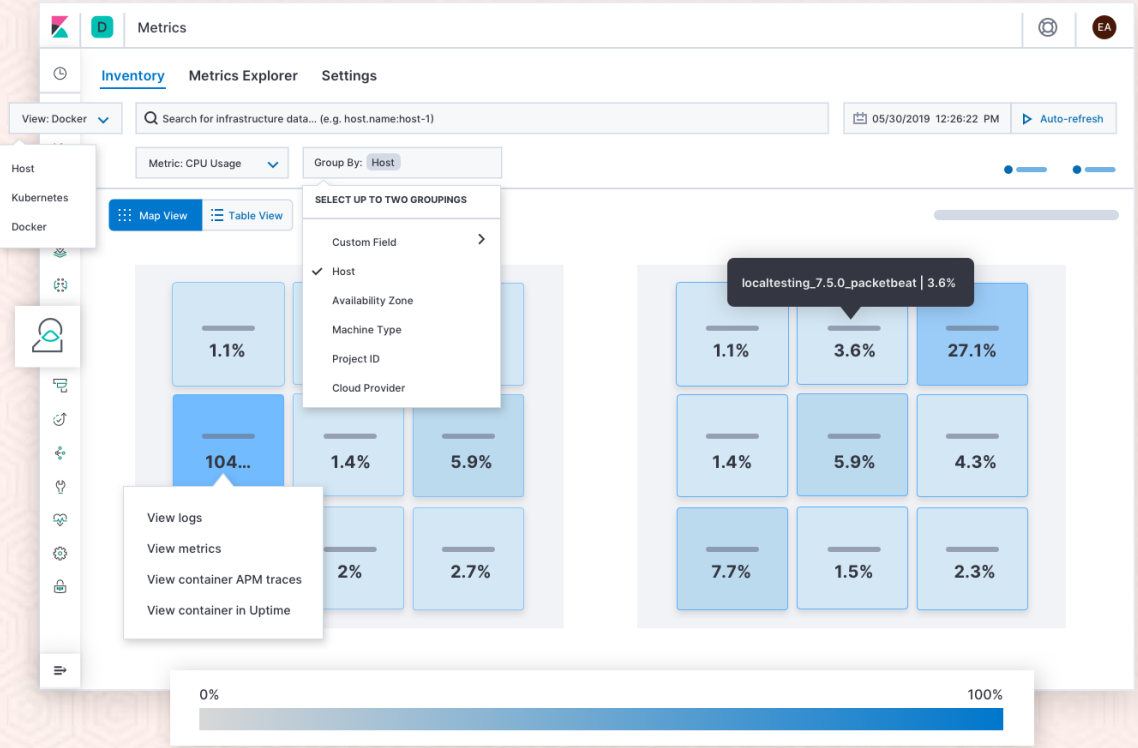
오늘의 학습내용

- Data Dashboard Workflow
- Beats를 활용한 실시간 로그관제 시스템 구축 실습
- ELK 활용 인구분석 시스템 구축 실습

Data Dashboard Workflow

오픈 소스 인프라 모니터링

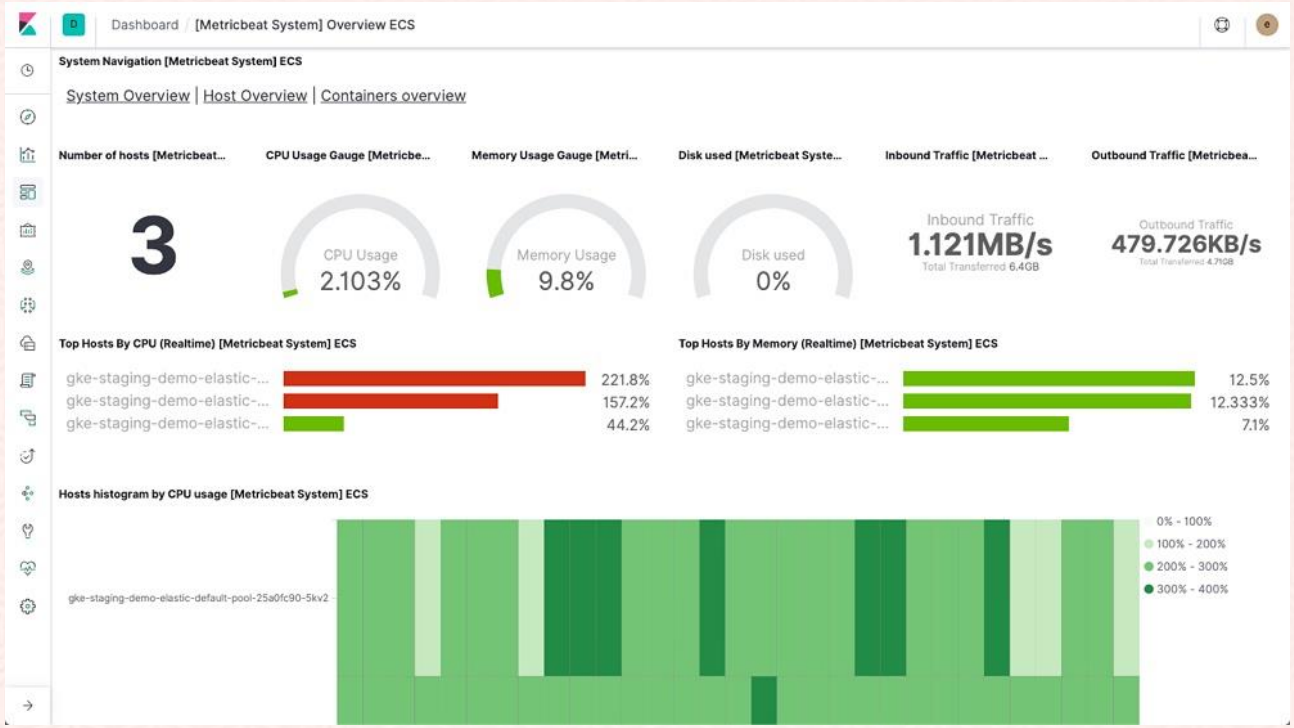
❖ 인프라 로그 및 메트릭 추적



Data Dashboard Workflow

오픈 소스 인프라 모니터링

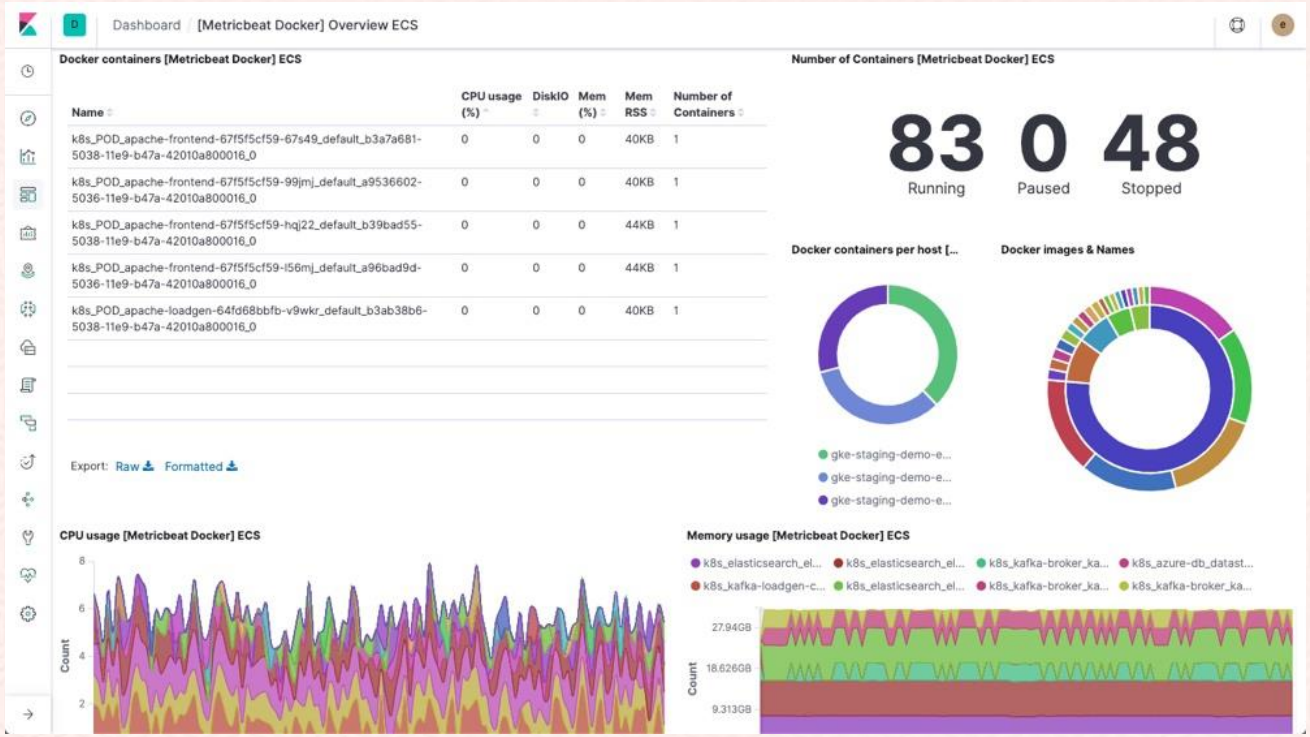
❖ **서버 : 인프라 로그 및 메트릭 추적**



Data Dashboard Workflow

오픈 소스 인프라 모니터링

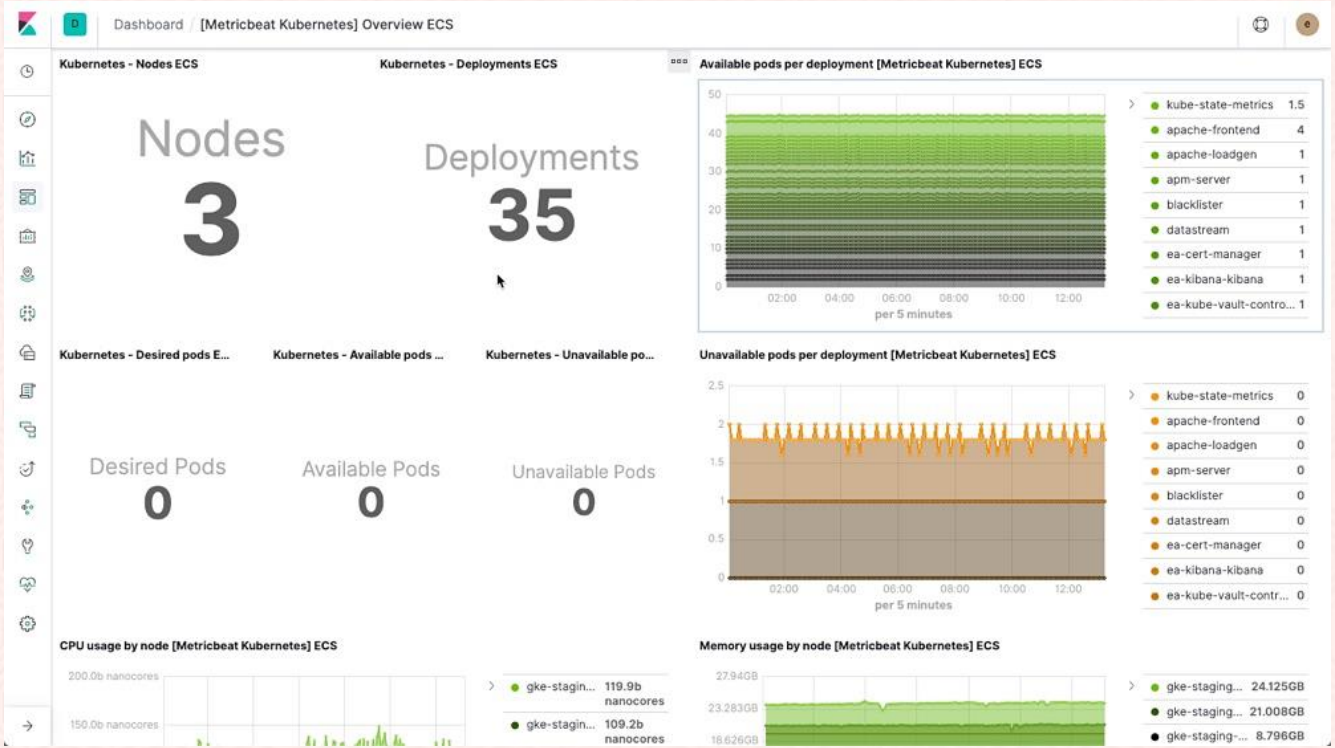
❖ Docker : 인프라 로그 및 메트릭 추적



Data Dashboard Workflow

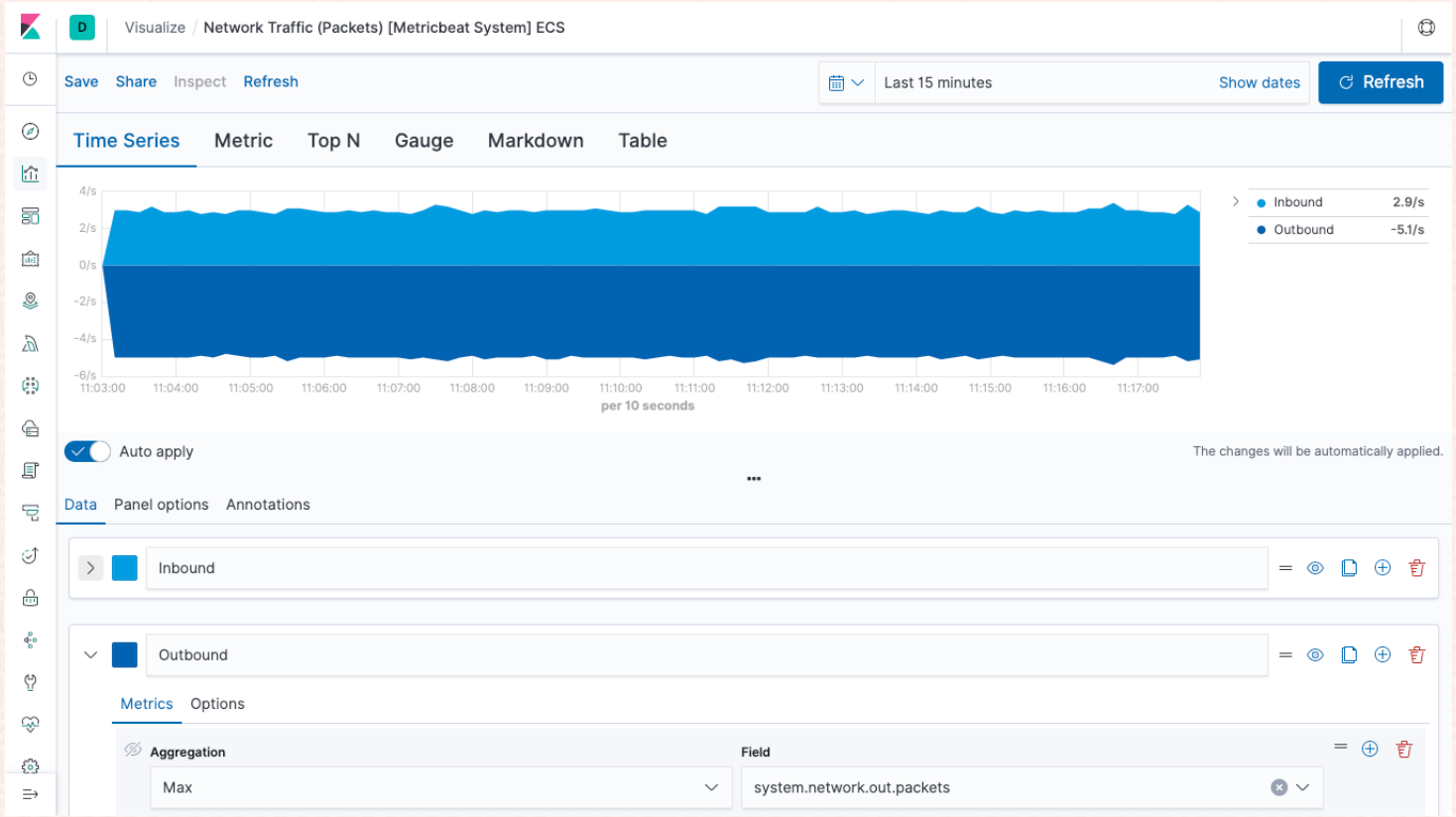
오픈 소스 인프라 모니터링

❖ **Kubernetes : 인프라 로그 및 메트릭 추적**



Data Dashboard Workflow

- 시계열 데이터를 위한 Elasticsearch

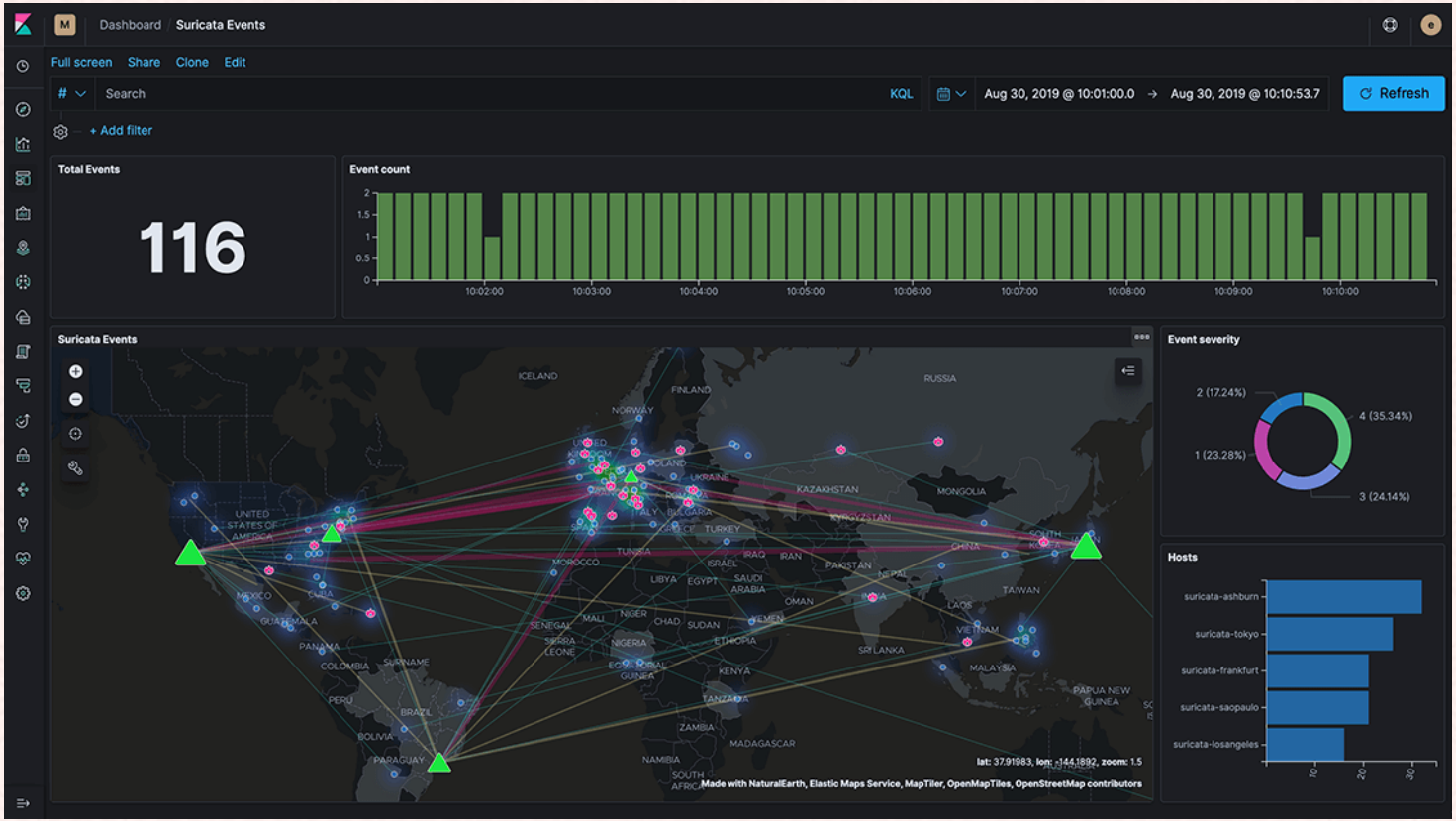


Data Dashboard Workflow

- 머신 러닝으로 이상 징후 탐색

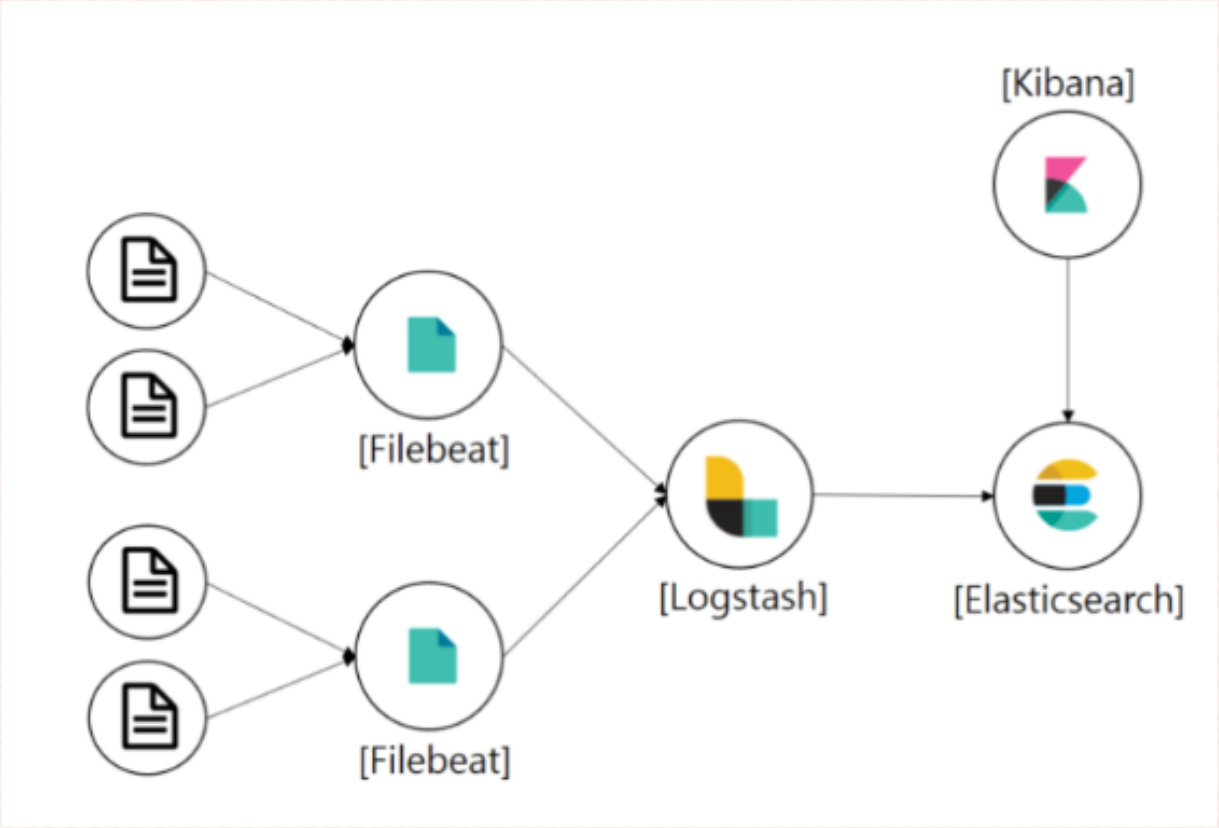
Data Dashboard Workflow

- 지도에 위치 정보 표시



Beats를 활용한 실시간 로그관제 시스템 구축 실습

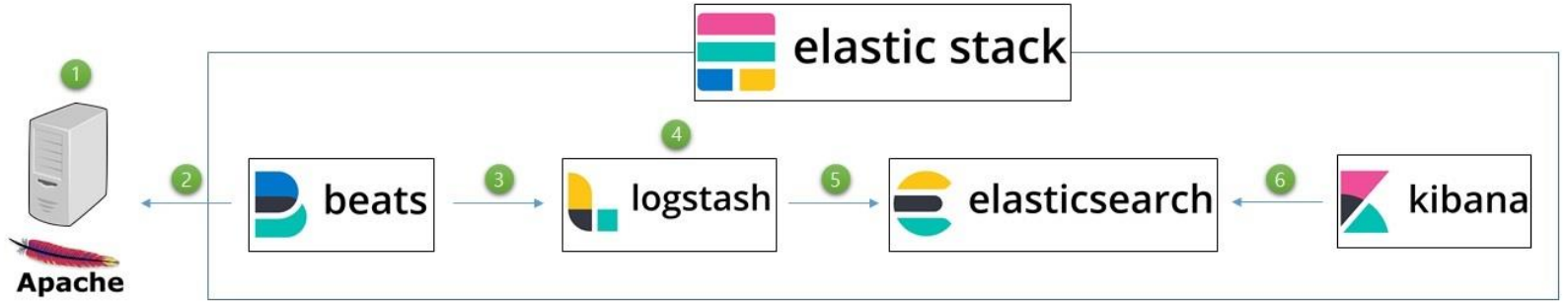
- ELK 스택 로그 수집 프로세스 구성



Beats를 활용한 실시간 로그관제 시스템 구축 실습

ELK 스택 로그 수집 프로세스 구성

- Elasticsearch : Apache Lucene 기반의 실시간 분산형 RESTful 검색 및 분석 엔진
- Logstash : 각종 로그를 수집하여 JSON 형태로 만들어 Elasticsearch로 데이터를 전송
- Kibana : Elasticsearch에 저장된 데이터를 사용자에게 Dashboard 형태로 보여주는 시각화 솔루션
- Filebeat : 로그 파일을 경량화시켜서 Elasticsearch 또는 Logstash로 전송하는 역할 수행. Logstash가 과부하 상태이면 전송하는 속도를 조절하고, 시스템 가동이 중단 혹은 재부팅되면 로그의 중단점을 기억하고 그 지점부터 다시 전송



Beats를 활용한 실시간 로그관제 시스템 구축 실습

환경 구축 및 수집기 구축

❖ JDK 설치

- OpenJDK 1.8.0_222 설치

- JAVA_HOME 및 PATH 변수 설정 : /etc/profile

❖ filebeat 설치 및 설정

$ wget https://artifacts.elastic.co/downloads/beats/filebeat/filebeat-6.6.0-linux-x86_64.tar.gz

$ tar -zxvf filebeat-6.6.0-linux-x86_64.tar.gz

$ ln -s filebeat-6.6.0-linux-x86_64 filebeat

❖ Filebeat 실행

$ ./filebeat -c access_log.yml -d publish &

Beats를 활용한 실시간 로그관제 시스템 구축 실습

환경 구축 및 수집기 구축

❖ filebeat 설정 : filebeat.yml

```
#==================== Filebeat inputs=======================

filebeat.inputs:

  - type: log

  enabled: true

  paths:- /var/log/*.log
```

Beats를 활용한 실시간 로그관제 시스템 구축 실습

환경 구축 및 수집기 구축

❖ filebeat 설정 : filebeat.yml

```
#=== ==============Outputs ======================
#------------------------- Elasticsearch output ------------------------------
#output.elasticsearch:
# Array of hosts to connect to.
#hosts: ["localhost:9200"]
#------------------------------ Logstash output -------------------------------
 output.logstash:
  hosts: ["localhost:5044"]
```

Beats를 활용한 실시간 로그관제 시스템 구축 실습

데이터 파이프라인 구축

❖ Logstash 설치 및 설정

```
$ wget https://artifacts.elastic.co/downloads/logstash/logstash-6.6.0.tar.gz

$ tar -zxvf logstash-6.6.0.tar.gz

$ ln -s logstash-6.6.0 logstash
```

❖ Logstash 실행

```
$ bin/logstash -f config/access_log.conf &
```

Beats를 활용한 실시간 로그관제 시스템 구축 실습

데이터 파이프라인 구축

❖ Logstash 설정 : config/logstash.conf

```
 # beats 에서 5044 port 로 데이터를 input 받겠다는 의미
input {
        beats {
                port => "5044"
        }
}
```

Beats를 활용한 실시간 로그관제 시스템 구축 실습

데이터 파이프라인 구축

❖ Logstash 설정 : config/logstash.conf

```
# grok 필터를 활용하여 엑세스 로그 한 줄을 아래처럼 파싱하겠다는 의미

# 해당 필터는 apache의 로깅 설정에 의해 만들어지는 파일의 포멧에 맞추어 설정해야한다.

filter {

        grok {

                match => { "message" => ["%{IPORHOST:clientip} (?:-|%{USER:ident}) (?:-
|%{USER:auth}) \[%{HTTPDATE:timestamp}\] \"(?:%{WORD:httpMethod} %{NOTSPACE:uri}(?:
HTTP/%{NUMBER:httpversion})?|-)\" %{NUMBER:responseCode} (?:-|%{NUMBER:bytes}) (?:-
|%{NUMBER:bytes2})( \"%{DATA:referrer}\")?( \"%{DATA:user-agent}\")?"] }

                remove_field => ["timestamp","@version","path","tags","httpversion","bytes2"]

        } ..
```

Beats를 활용한 실시간 로그관제 시스템 구축 실습

데이터 파이프라인 구축

❖ Logstash 설정 : config/logstash.conf

```
# 정제된 데이터를 elasticsearch 에 인덱싱

# index 이름에 날짜 형태로 작성되면 인덱싱 시점의 시간에 따라 인덱싱 이름이 자동으로 변경(아래는 월별로 인덱스를 만들 경우)

output {
        elasticsearch {
                hosts => [ "{elasticsearch ip}:9200" ]
                index => "index-%{+YYYY.MM}"
        }
}
```

Beats를 활용한 실시간 로그관제 시스템 구축 실습

Elasticsearch

❖ Elasticsearch 설치 및 구축

```
$ wget https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-6.6.0.tar.gz
$ tar zxvf elasticsearch-6.6.0.tar.gz
$ ln -s elasticsearch-6.6.0 elasticsearch
```

❖ Elasticsearch 실행

```
$ cd ../bin
$ echo './elasticsearch -d -p es.pid' > start.sh
$ echo 'kill `cat es.pid`' > stop.sh
$ chmod 755 start.sh stop.sh
```

Beats를 활용한 실시간 로그관제 시스템 구축 실습

시각화

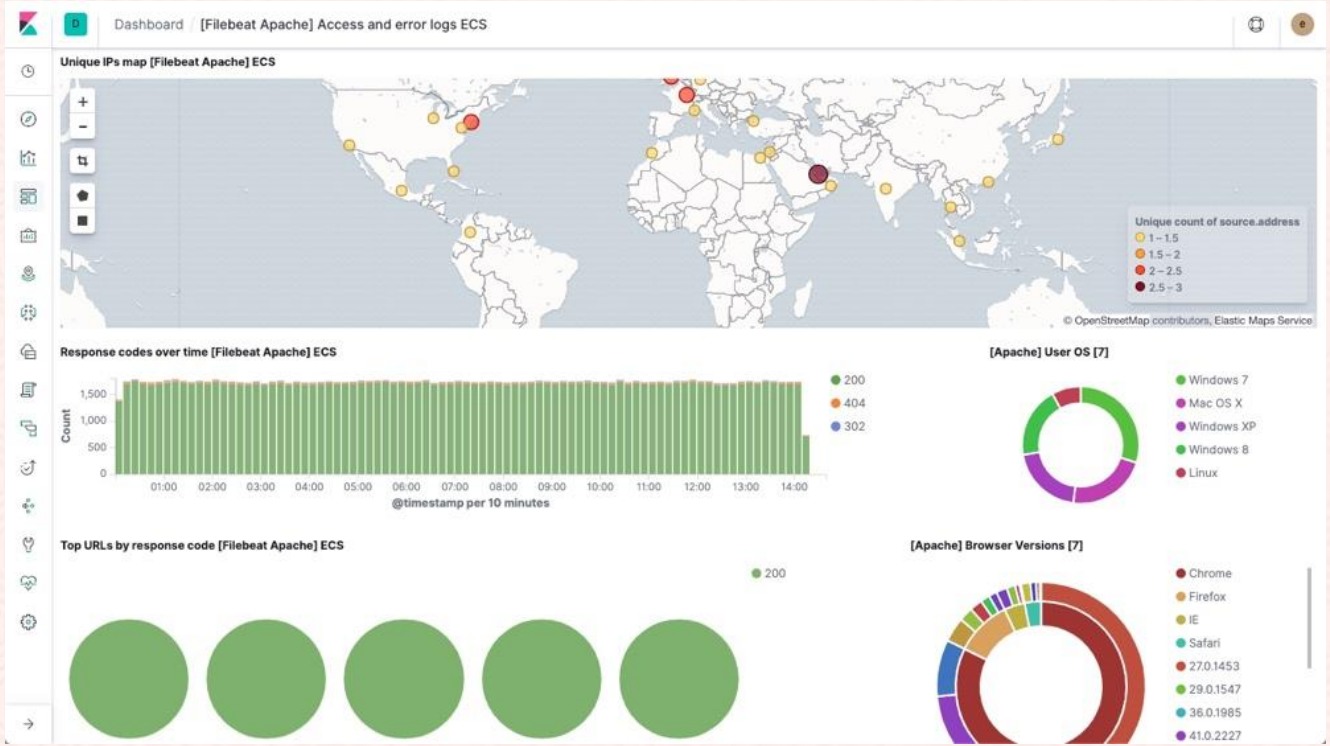
❖ Kibana 설치 및 구축

```
$ https://artifacts.elastic.co/downloads/kibana/kibana-6.6.0-linux-x86_64.tar.gz
$ tar -zxvf kibana-6.6.0-linux-x86_64.tar.gz
$ ln -s kibana-6.6.0-linux-x86_64 kibana
$ cd kibana/config
$ vi kibana.yml
   server.host: "@.@.@.@"
   elasticsearch.hosts: ["http://@.@.@.@:9200"]
$ cd bin/
$ nohup ./kibana &
```

Beats를 활용한 실시간 로그관제 시스템 구축 실습

시각화

❖ Kibana Apache 로그



ELK 활용 인구분석 시스템 구축 실습

인구 분석

- Dataset : http://catalog.data.gov/dataset
- Population by country download
- Kibana & Elasitcsearch 실행 여부 확인 ps -ef | grep
- Logstash 구축

```
input {
  file {
    path => "/home/linux/populationbycountry19802010millions.csv" ## 절대 경로
    start_position => "beginning" ## end가 기본설정이다 하지만 파일에서 직접 받을 거기 때문에 처음부터 받는다. Streaming data는 END (default)
    sincedb_path => "/dev/null" ## 이걸 넣지 않는다면 한번 데이터는 다시 넣지 않는다.
  }..
```

ELK 활용 인구분석 시스템 구축 실습

인구 분석

```
filter {
  csv {
      separator => "," ## 구분자
      columns => ["Country","1980","1981","1982","1983","1984","1985","1986","1987‟,…,”2010”]
  }
  mutate {convert => ["1980", "float"]}
  mutate {convert => ["1981", "float"]}
  …
  mutate {convert => ["2009", "float"]}
  mutate {convert => ["2010", "float"]}
}
```

ELK 활용 인구분석 시스템 구축 실습

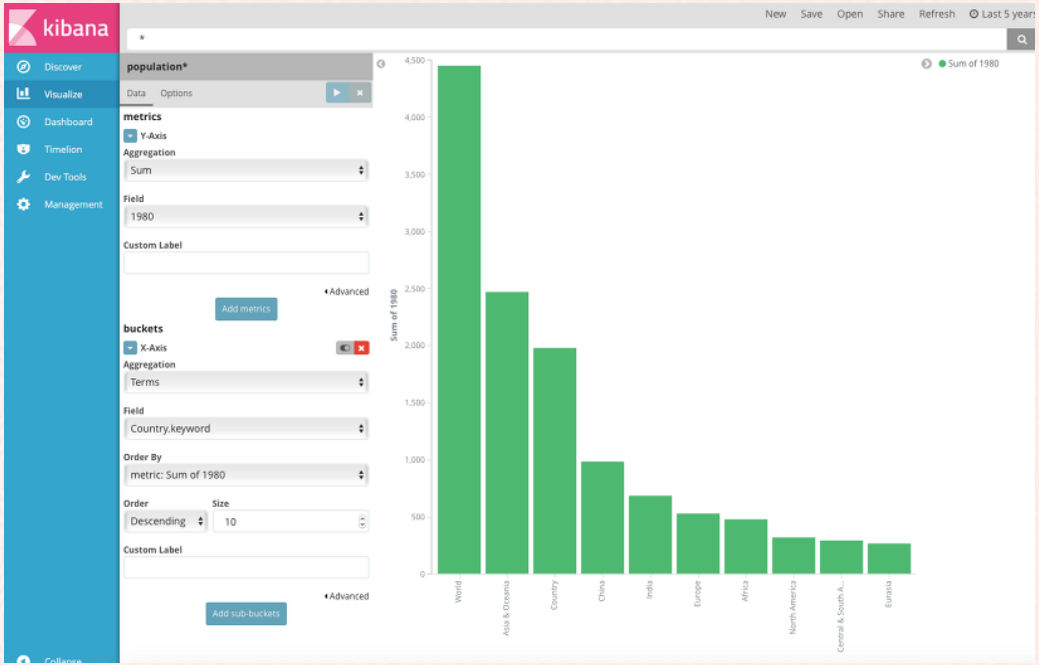
인구 분석

```
output {
    elasticsearch {
        hosts => "localhost"
        index => "population" # index
    }
    stdout {} # 로그 확인.
}
```

ELK 활용 인구분석 시스템 구축 실습

시각화

- ❖ Discovery 에서 원하는 필드를 선택하고 원하는 값들을 다양하게 Filter 를 통해 확인 가능
- ❖ Visualize 를 통해 시각화



ELK 활용 인구분석 시스템 구축 실습

시각화

❖ Discovery 에서 원하는 필드를 선택하고 원하는 값들을 다양하게 Filter 를 통해 확인 가능

❖ 대시보드