프로젝트 결과 보고서

작 품 명	OpenAPI 시각화로 분석하는 공공 데이터
팀 명	놀면 뭐 하니
제 작 기 간	2019년 8월 7일 ~ 2018년 8월 14일
프로젝트 매니저	최민영 (인)

조 원			
No.	성 명	연 락 처	E-mail
1	최민영	010-2994-5914	asakura14@naver.com
2	박태수	010-6260-8662	pts6260@gmail.com
3	노희욱	010-9412-7043	blashi@naver.com
4	김정인	010-8344-7529	jigijigi0421@naver.com
5	정우민	010-5191-5335	post4482@naver.com

● 조 소개

➢ 팀명

동명의 TV 프로그램 '놀면 뭐 하니?'를 인용했습니다. 데이터 놔두고 놀면 뭐하니?

➢ 역할

비전공자들이 많아 수업 시간에 진행한 ELK 환경 구축에 이해를 위해 조원 모두 단계별로 함께 진행하였습니다. 특별히 역활을 나누고 진행하지 않았으나, 참여도가 높았던 일 감별로 분류합니다.

- OpenAPI 연동, 결과 보고서 작성: 최민영
- ELK 환경 구성: 박태수, 정우민
- 공공 데이터 리서치: 김정인, 노희욱

- **프로젝트 주제**

  OpenAPI 시각화로 분석하는 공공 데이터.

- **프로젝트 개요**

  ➢ **프로젝트 소개**

351 systems in ranking, August 2019

Rank			DBMS	Database Model	Score		
Aug 2019	Jul 2019	Aug 2018			Aug 2019	Jul 2019	Aug 2018
1.	1.	1.	Oracle	Relational, Multi-model	1339.48	+18.22	+27.45
2.	2.	2.	MySQL	Relational, Multi-model	1253.68	+24.16	+46.87
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model	1093.18	+2.35	+20.53
4.	4.	4.	PostgreSQL	Relational, Multi-model	481.33	-1.94	+63.83
5.	5.	5.	MongoDB	Document	404.57	-5.36	+53.59
6.	6.	6.	IBM Db2	Relational, Multi-model	172.95	-1.19	-8.89
7.	7.	↑ 8.	Elasticsearch	Search engine, Multi-model	149.08	+0.27	+10.97
8.	8.	↓ 7.	Redis	Key-value, Multi-model	144.08	-0.18	+5.51
9.	9.	9.	Microsoft Access	Relational	135.33	-1.98	+6.24
10.	10.	10.	Cassandra	Wide column	125.21	-1.80	+5.63

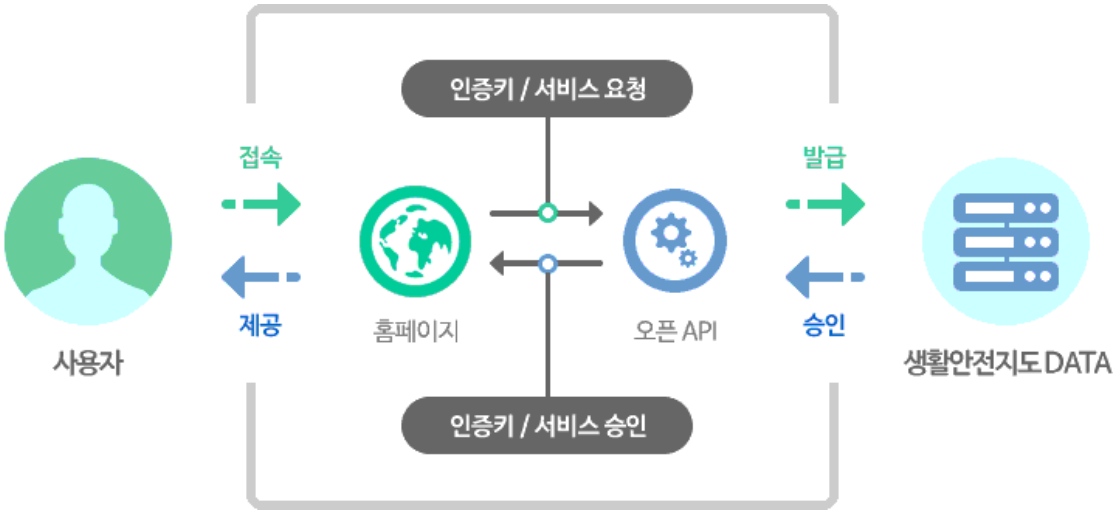
출저: https://db-engines.com/en/ranking

최근 Elasticsearch는 고전적인 DB 시장에서 매섭게 올라오고 있습니다. 'DB-ENGINES'에서 조사한 자료에 따르면, 2019년도에는 2018년도 보다 한단계 상승한 7위에 올라와 있습니다. 이는 Elasticsearch가 '빅데이터 시장과 함께 성장을 하고 있다' 라고 해석할 수 있습니다.

본 프로젝트에서는 빅데이터 분석 환경 경향에 맞추어, 공공 데이터분석을 위해 Elasticsearch와 함께 구성하는 ELK(Elasticsearch, Logstash, Kibana, ELK) 환경을 구축하고, 데이터 수집을 위해 서울시에서 제공하는 OpenAPI를 사용하며, 수집된 데이터를 Kibana로 시각화하여 분석합니다.

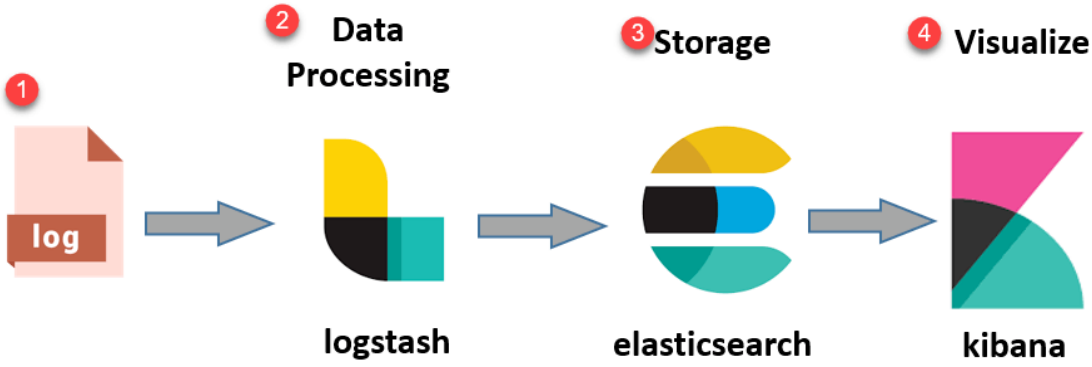
  ➢ **프로젝트 선정 이유**

빅데이터 분석에서 첫걸음은 분석에 사용할 질 좋은 데이터를 수집하는 것에서부터 시작합니다. 또한, 분석 성능에 개선을 위하여 일회성이 아닌 지속적인 데이터의 수집도 필요합니다. 따라서 위의 조건을 만족하고자, 공공 데이터 및 서울시에서 제공하는 OpenAPI를 선택했습니다.



출저: http://www.safemap.go.kr/dvct/openAPI.do

OpenAPI는 인터넷 이용자가 일방적으로 웹 검색 결과 및 사용자 화면 등을 제공받는 데 그치지 않고 직접 응용 프로그램과 서비스를 개발할 수 있도록 공개된 개발자를 위한 인터페이스입니다. RESTFul API라고도 할 수 있으며, 서버가 클라이언트에게 제공하는 서비스를 URI로 접근할 수 있는 특징이 있습니다.



출저: https://cr4ft-ing.tistory.com/189

ELK 환경은 Elasticserach, Logstash, Kibana를 통틀어 지칭하는 용어입니다. Logstash는 데이터를 수집하고 정제하는 기능을 하고, Elasticsearch는 인덱싱이라는 기술을 사용하여 수집된 데이터를 저장하며, Kibana는 Elasticserch에 저장된 데이터를 시각화하여 분석을 할 수 있게 합니다. 여기에 FileBeat 라는 프로그램을 Logstash와 연동하면, 로컬 머신에서 생성하는 파일을 수집 할 수 있는 환경을 만들 수 있습니다.

위와 같은 기술적인 이유도 있지만, 다양한 전공으로 구성되어 있는 조원 모두가 빅데이터 분석에 첫걸음을 성공적으로 시작하고자 이와 같은 프로젝트를 선정했습니다.

● 활용 범위

➢ 데이터 시각화

Kibana는 강력한 시각화 도구를 제공합니다. 이를 활용하여 Elasticsearch가 저장하는 빅 데이터를 시각화하여 데이터를 해석하고, 새로운 분석 모델을 만드는데 도움을 줍니다.

➢ 공공 데이터 분석

공공 데이터 포털(https://www.data.go.kr/)에서 제공하는 데이터는 기본적으로 OpenAPI로 접근이 가능합니다. OpenAPI가 서버로부터 받아오는 데이터의 형태(e.g. Jason, XML etc.)는 다를 지라도 수집 방법은 동일합니다. 따라서 본 프로젝트에서 수행한 환경을 구축하면 접근 가능한 모든 공공 데이터를 수집할 수 있습니다.

➢ IoT 데이터 분석

FileBeat라는 프로그램은 자신이 설치되어있는 로컬 머신에 파일 데이터를 수집합니다. 또한, 본 프로그램은 Logstash와 연동 가능합니다. 이는 엣지 컴퓨팅(Edge Computing)을 수행하는 IoT 머신이 지속적으로 생성하는 계측 파일과 로그 파일을 ELK환경을 이용하여 분석할 수 있다는 것을 의미합니다.

● 프로젝트 내용

➢ ELK 환경 구성 및 설치

ELK가 설치 될 머신은 리눅스 OS를 사용한다고 가정합니다. 본 문서에서는 CentOS 7(Minimal)으로 진행합니다. 원활한 진행을 위해서 네트워크는 고정 아이피를 사용하는 브릿지 모드로 동작합니다. 또한 CentOS에 루트 계정을 사용하지 말고, 신규 계정을 생성해서 진행하도록 합니다.

1. Java 1.8 설치

ELK Stack이 동작하려면 Java가 설치 되어야 합니다. CentOS 7을 Minimal로 설치 했다면 바로 설치하면 되지만, 그렇지 않다면 기존의 Java를 삭제하도록 합니다. 아래에 명령어를 수행합니다.

$ java –version

```
[ec2-user@ip-                ~]$ java -version
java version "1.7.0_151"
OpenJDK Runtime Environment (amzn-2.6.11.0.74.amzn1-x86_64 u151-b00)
OpenJDK 64-Bit Server VM (build 24.151-b00, mixed mode)
[ec2-user@ip-                ~]$
```

결과에 나온 버전을 삭제하도록 합니다. 아래에 명령어를 수행합니다.

$ yum remove java-1.7.0-openjdk.x86_64 –y

이후, 본 설치 환경에 맞는 Java를 설치합니다. 아래에 명령어를 수행합니다.

$ yum install java-1.8.0-openjdk-devel.x86_64 –y

설치 후, 아래와 같은 자바 버전을 확인 할 수 있습니다.

```
[ec2-user@ip-172-31-12-244 ~]$ java -version
openjdk version "1.8.0_151"
OpenJDK Runtime Environment (build 1.8.0_151-b12)
OpenJDK 64-Bit Server VM (build 25.151-b12, mixed mode)
[ec2-user@ip-172-31-12-244 ~]$
```

2. Elasticserach 설치 및 설정

본 설치 과정에서는 ELK Stack을 하나의 디렉토리에 모아둘 것입니다. 원하는 디렉토리에서 다음에 명령어를 수행합니다.

$ mkdir elk

$ cd elk

이후 wget을 사용하여 elk 디렉토리에서 Elasticsearch를 다운로드합니다. 다음에 명령어를 수행합니다.

$ wget https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-5.6.4.tar.gz

$ tar -xzvf elasticsearch-5.6.4.tar.gz

$ rm elasticsearch-5.6.4.tar.gz

이것으로 Elasticsearch 설치를 성공했습니다. 다음에는 환경설정입니다.

- Limits.conf 편집

  Elasticsearch가 정상 작동하려면 파일 디스크립터를 65536까지 올려야 합니다. 이를 위해 다음을 수행합니다.

  $ ulimit –Hn

  별다른 설정을 안했다면 4096이 출력될 것입니다. 다음으로,

  $ vi /etc/security/limits.conf

  Vi 편집기를 사용하여 limits.conf 파일을 수정합니다. 가장 아래에 다음에 두 줄을 추가하도록 합니다.

```
*          hard     nofile          65536
*          soft     nofile          65536
```

  Esc를 누른 후, wq를 입력하여 저장을 한 뒤 vi를 종료합니다. 그리고 반드시 재접속을 합니다. 그 뒤, 위에서 언급한 명령어로 파일 디스크립터가 수정되었는지 확인합니다.

- 가상 메모리 영역 늘리기

  Elasticsearch는 mmapfs 디렉토리에 기본 설정으로 index를 저장합니다. mmap counts에 대한 운영체제의 limit이 default가 낮게 설정되어서 올려주지 않으면 'out of memory' 에러가 발생합니다.

  $ vi /etc/sysctl.conf

  열린 파일에 아래를 입력합니다.

vm.max_map_count=262144

편집기를 저장하고 종료한 뒤, 재접속을 합니다.

- JVM 설정

  본 설정은 머신(혹은 가상 머신)에 메모리 할당량에 따라 달라집니다. 머신 시스템이 허용 가능한 최적의 메모리를 선택해주면 됩니다. 방법은 아래와 같습니다.

  $ cd write_here_your_dir/elk/elasticsearch-5.6.4

  $ vi ./config/jvm.options

  minimum heap size (Xms)와 maximum heap size (Xmx) 가 동일하게 설정할 것을 권장합니다. 예는 아래와 같습니다.

```
-Xms4G
-Xmx4G
```

- config 설정

  $ cd write_here_your_dir/elk/elasticsearch-5.6.4

  $ vi ./config/elasticsearch.yml

  다음에 항목을 설정 혹은 추가합니다.

```
network.host: your static ip
discovery.type: single-node
```

  Elasticsearch는 DB 서버로 동작하기 위해 호스트 아이피가 필요합니다. 또한, localhost로 동작하지 않는 Elasticsearch는 반드시 클러스터링 노드 설정을 해주어야 합니다. 방법은 설정 파일에 주석으로 'Discovery'를 찾아 해당 부분을 설정해주면 됩니다. 기본적으로 제공되는 설정파일에는 모든 설정 옵션들이 적혀 있지 않습니다. 따라서 위처럼 클러스터링을 싱글 노드로 구성하려면 해당 줄을 추가해야 합니다.

3. Logstash 설치 및 설정

아래에 명령어를 사용해 설치를 합니다.

$ cd write_here_your_dir/elk

$ wget https://artifacts.elastic.co/downloads/logstash/logstash-5.6.4.tar.gz

$ tar -xzvf logstash-5.6.4.tar.gz

$ rm logstash-5.6.4.tar.gz

이것으로 Logstash 설치를 성공했습니다. 다음에는 환경 설정을 합니다.

- JVM 설정

  Elasticsearch와 동일합니다. 설정 파일(logstash-5.6.4/config/jvm.options)에 옵션도 동일하므로 생략합니다.

- config 설정

  Logstash에 config는 아직 설정할 필요가 없습니다. 필요하다면 공식 홈페이지에서 해당 사항을 참조하도록 합시다. 다만 프로그램을 실행할 때 사용할 '.conf' 파일을 생성해야 합니다. 해당 사항은 이 후 'Logstash – OpenAPI 환경 설정'에서 자세히 설명합니다.

4. Kibana 설치 및 설정

아래에 명령어를 사용해 설치를 합니다.

$ cd write_here_your_dir/elk

$ wget https://artifacts.elastic.co/downloads/kibana/kibana-5.6.4-linux-x86_64.tar.gz

$ tar -xzvf kibana-5.6.4-linux-x86_64.tar.gz

$ rm kibana-5.6.4-linux-x86_64.tar.gz

이것으로 Kibana 설치를 성공했습니다. 다음에는 환경 설정을 합니다.

- **config 설정**

  $ cd write_here_your_dir/elk/kibana-5.6.4-linux-x86_64

  $ vi ./config/kibana.yml

  Kibana는 웹 서버로 동작하기 위해 호스트 아이피가 필요합니다. 또한, Elasticsearch가 인덱싱한 데이터를 사용하고자 Elasticsearch에 주소가 필요합니다.

  아래에 항목을 찾아 설정하도록 합니다. 없으면 해당 줄을 추가합니다.

```
server.host: "your static ip"
elasicsearch.url: "http://your static ip:9200"
```

5. FileBeat 설치 및 설정

아래에 명령어를 사용해 설치를 합니다.

$ cd write_here_your_dir/elk

$ wget https://artifacts.elastic.co/downloads/beats/filebeat/filebeat-7.3.0-linux-x86_64.tar.gz

$ tar -xzvf filebeat-7.3.0-linux-x86_64.tar.gz

$ rm filebeat-7.3.0-linux-x86_64.tar.gz

이것으로 FileBeat 설치를 성공했습니다. 다음에는 환경 설정을 합니다.

- config 설정

  $ cd write_here_your_dir/elk/filebeat-7.3.0-linux-x86_64

  $ vi ./filebeat.yml

  FileBeat는 특이하게 config 디렉토리가 따로 없습니다. 편집기로 바로 열어 아래를 수정하도록 합니다.

```
Filebeat.input:
-   type: log
      enabled: true
      paths:
        - your log file path here
```

```
Output.logstash:
  Hosts: "your static ip:5044"
```

  만약 Logstash를 제외한 나머지 연동 프로그램이 Output에 활성화 되어있으면 FileBeat는 작동하지 않습니다. 반드시 하나의 Output만을 활성화 시키도록 합니다.

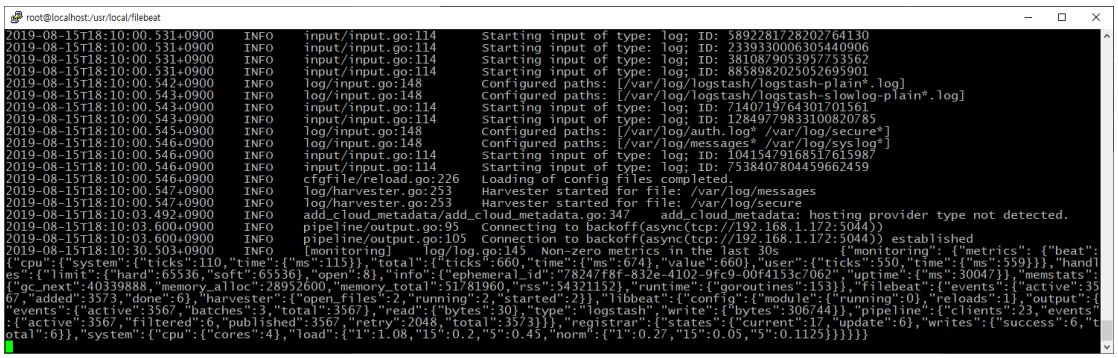
- 실행 확인

  본 프로젝트에서 따로 FileBeat를 사용하지 않아 본 항목에 실행 방법을 서술합니다. 아래에 커맨드를 실행합니다.

  $ cd write_here_your_dir/elk/filebeat-7.3.0-linux-x86_64

  $ ./filebeat –e ./filebeat.yml –c

실행하면 다음에 화면을 확인 할 수 있습니다.

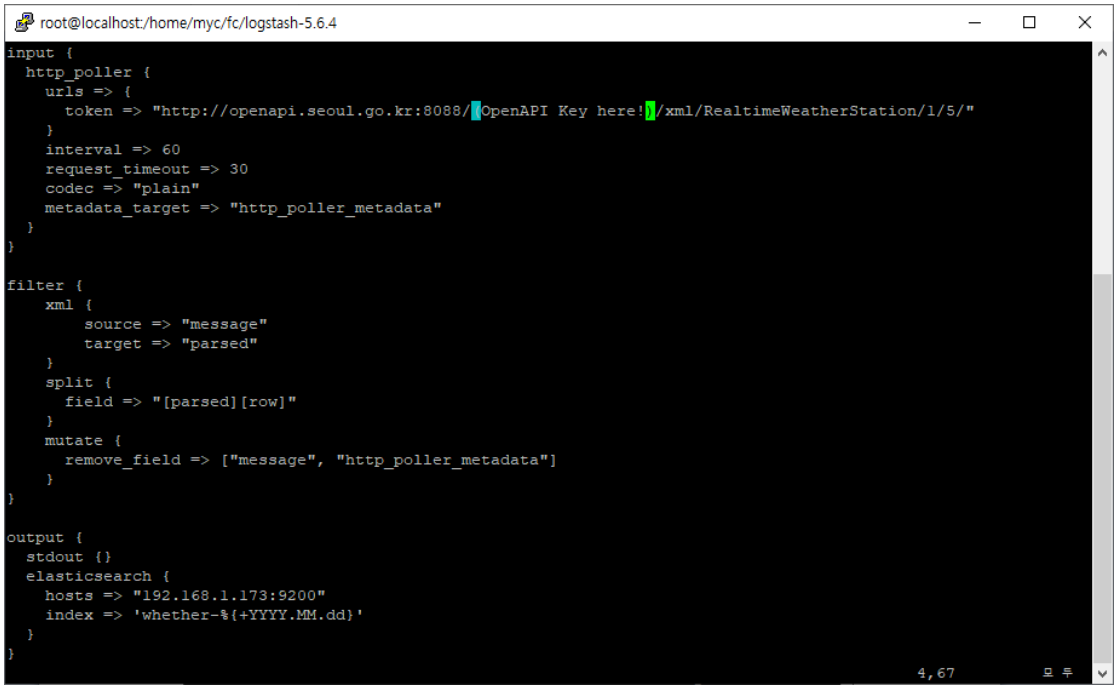


➢ Logstash - OpenAPI 연동

- OpenAPI

  OpenAPI를 활용하려면 인증키를 발급 받아야 합니다. 본 프로젝트를 수행하고자 서울 열린 데이터 광장(http://data.seoul.go.kr)에 가입하고 인증키를 발급 받았습니다. 자세한 사항은 해당 홈페이지를 참조하도록 합니다. 인증키를 발급받으면 OpenAPI를 사용할 준비는 완료되었습니다.

- Logstash - .conf 파일 생성



$ cd write_here_your_dir/elk/logstash-5.6.4

$ cd ./config

$ vi data.conf

위 명령어를 실행 해, 'data.conf' 파일을 생성한 뒤, 위 그림처럼 파일을 수정합니다. 'input'은 Logstash에 입력 부분을 설정합니다. 'http_poller'는 Logstash에서 지원하는 폴링 클라이언트 입니다. 'urls'에 폴링할 서버 주소는 적습니다. 여기서는 OpenAPI 주소가 됩니다. 중요한 점은 '(OpenAPI key here!)' 부분에 발급받은 인증키를 넣어야 합니다.

'filter'는 Logstash에 강력한 기능인 '정제'를 설정합니다. 여기서는 OpenAPI에 응답이 XML 데이터이기 때문에, XML을 필터링하는 설정을 합니다.

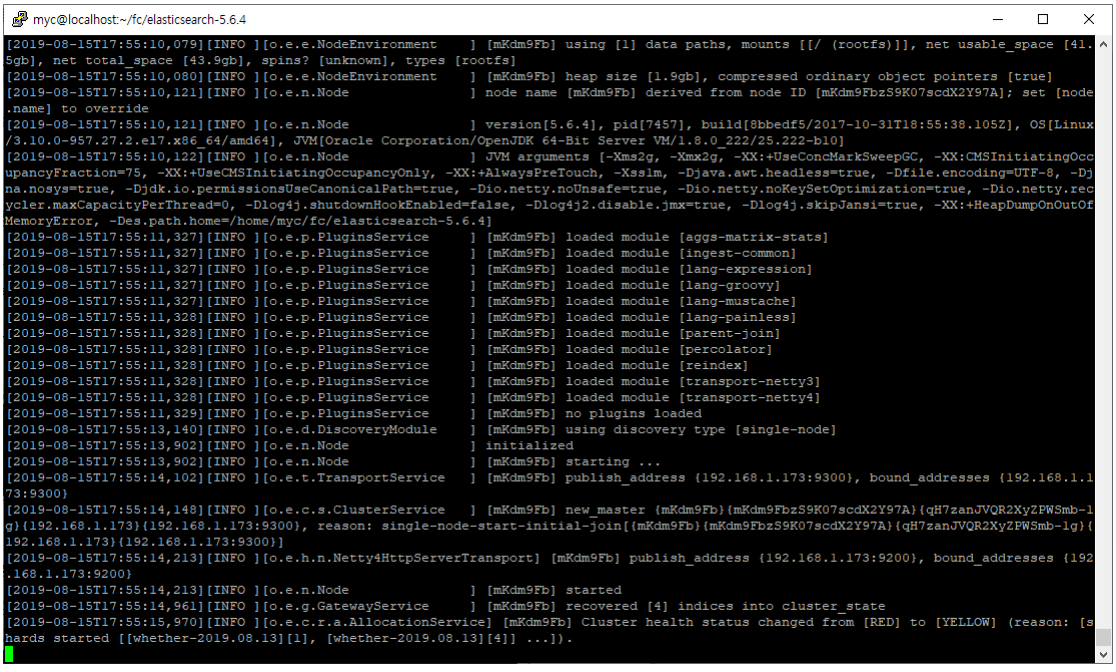
'output'은 Logstash가 정제한 데이터를 전달할 출력 부분을 설정합니다. 'stdout'은 표준 출력으로 디버깅 용도로 사용할 수 있습니다. 'elasticserach'에서 위에서 설정한 주소를 설정하고, 반드시 'index'를 설정합니다. 이 부분을 설정하지 않으면 Kibana가 패턴을 인식하지 못해 데이터를 분석할 수 가 없습니다. 또한 index가 'wheter-'로 시작하는 것을 기억합시다.

■ ELK Stack 실행

ELK Stack에 실행 순서는 중요합니다. 만약 Logstash에 Output에 연결이 활성화 되지 않으면 Logstash는 실행 되지 않습니다. 또한 Elasticsearch가 작동되지 않은 채로 Kibana를 실행하면 Kibana는 Elasticsearch를 계속 찾는 에러를 출력합니다.

위에 설정을 성공적으로 수행하였다면, 다음에는 순서대로 프로그램을 실행하면 데이터를 수집하고, 정제하며, 시각화를 할 수 있습니다.

1. Elasticsearch 실행

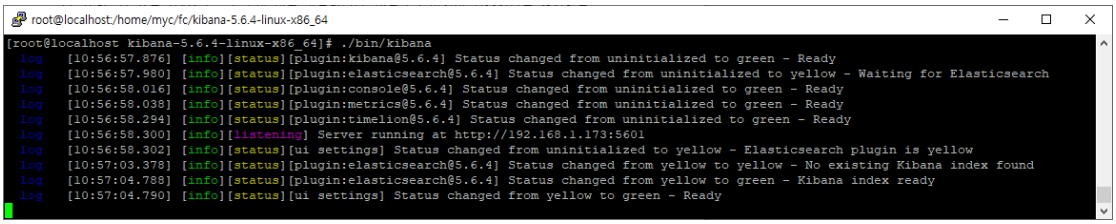


가장 먼저 Elasticsearch를 실행합니다. 설치 디렉토리에서 다음을 입력합니다.

$ ./bin/elasticserch

버전에 따라 root 유저가 실행 할 수 없으므로, 미리 생성한 계정으로 접속하여 실행합니다. 실행하게 되면 위와 같은 화면을 볼 수 있습니다.

2. Kibana 실행

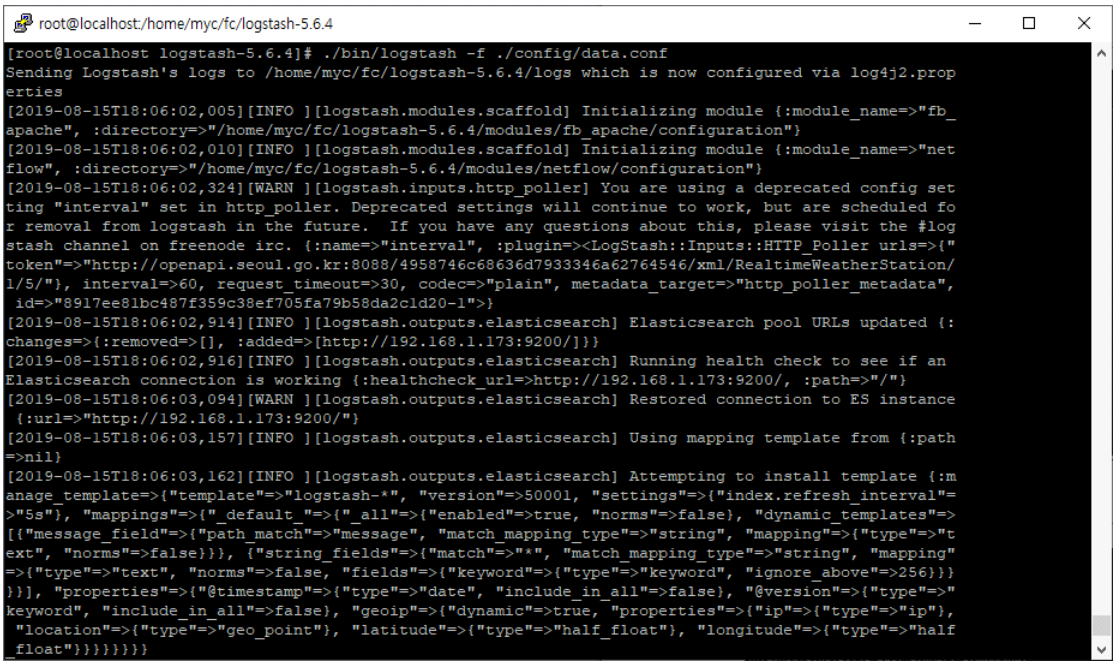


Kibana에 실행은 간단합니다. 설치 디렉토리로 이동하여 아래에 커맨드를 실행합니다.

$ ./bin/kibana

이후 위와 같은 화면을 확인 할 수 있습니다.

3. Logstash 실행



설치 디렉토리로 이동하여 아래에 커맨드를 수행합니다.

$ ./bin/logstash -f ./config/data.conf

이후 위와 같은 화면을 확인 할 수 있습니다.

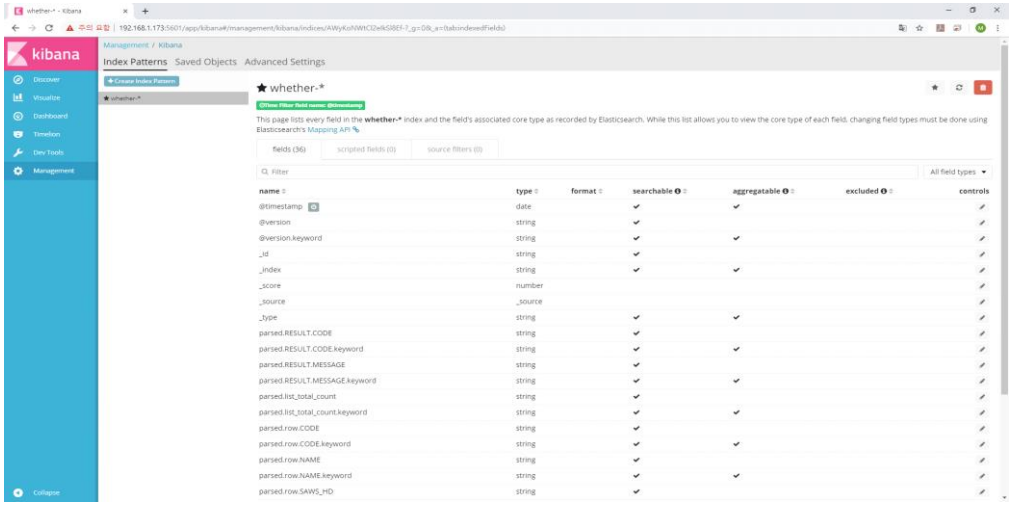
- **Kibana 시각화**

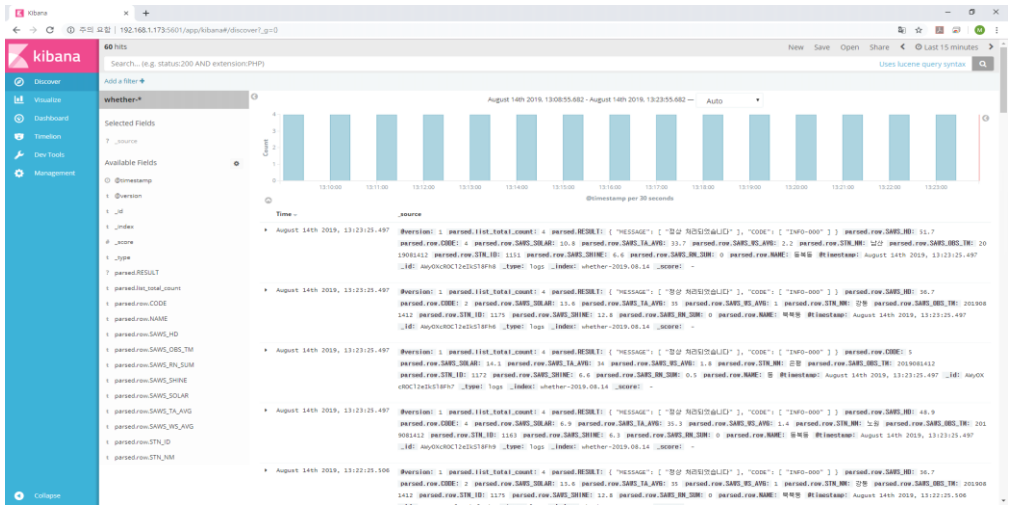
위에서 성공적으로 ELK Stack을 실행 하였다면 Logstash가 서울 열린 데이터 광장 서버로부터 1분마다 데이터를 받아와 Elasticserch로 보내고 있을 것입니다.

현재 머신에 네트워크가 브릿지로 구성되어 있으므로, 같은 네트워크에 접속할 수 있는 컴퓨터라면 호스트쪽에서 게스트 머신으로 접속할 수 있습니다. 단, 방화벽을 해재해야 합니다.

Kibana에 접속은 'http://고정아이피:5601'로 접속할 수 있습니다. 'https'가 아닌 것을 확인하세요. 현재 설정은 보안 접속을 설정하지 않아 'https'로는 접속을 할 수 없습니다.

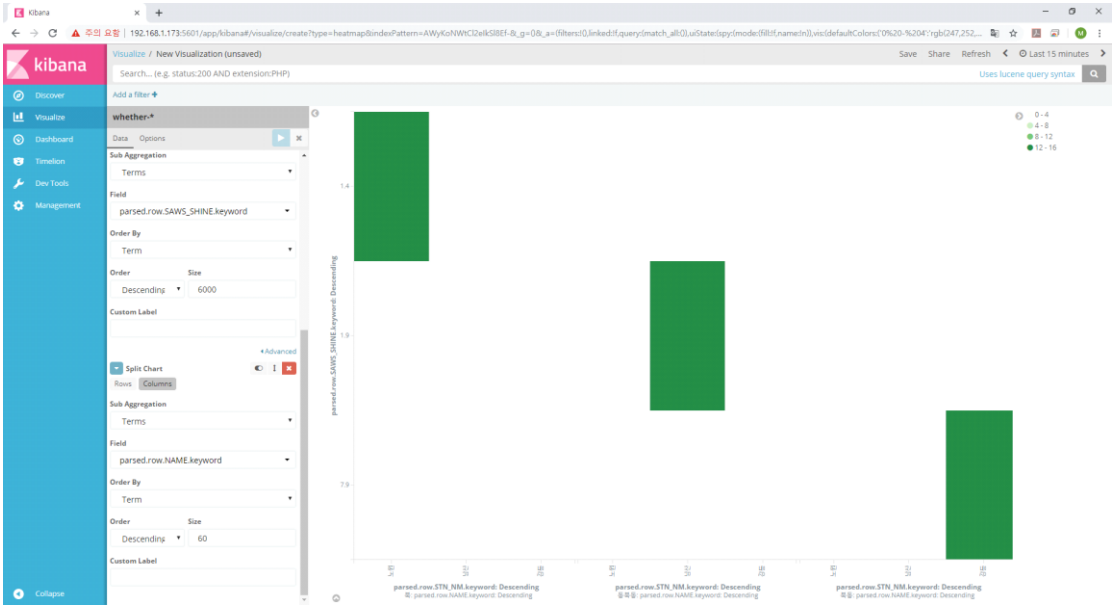
접속 후, 데이터를 확인하려면 Pattern을 등록해야 합니다. 왼쪽 메뉴에서 'Management->Index Patterns'를 선택하고 'whether-*' 패턴을 등록합니다. 만약 Logstash에서 Index 설정을 해주지 않았다면, 여기서 패턴을 등록할 수 없습니다. 반드시 확인하도록 합니다. 설정이 완료되면 아래에 화면을 확인할 수 있습니다.





이제 'Discovery' 메뉴에서 성공적으로 로그가 인덱싱 된 것을 확인 할 수 있습니다. Logstash에 http_poller가 1분 마다 폴링을 하기 때문에 인덱싱도 1분마다 처리되는 것을 확인 할 수 있습니다.

다음에는 위 데이터를 시각화를 합니다, 'Visualize' 메뉴를 선택하여 원하는 도구를 선택합니다. 여기서는 'Heat Map'을 선택했습니다. 그래프에 설정값을 원하는 방식으로 설정해 주면 아래와 같은 화면을 볼 수 있습니다. 사용한 OpenAPI는 서울시(노원, 남산, 강동)에 날씨 데이터를 가져오는 API 였고, 이것으로 날씨 변화에 패턴을 눈으로 확인해 볼 수 있습니다. 아래에 그래프를 해석해보면 노원에 일사량이 가장 많으며, 강동에 일사량이 가장 적음을 확인할 수 있고, 노원, 남산, 강동 세 곳에 일사량에 관계는 선형일 것이라는 추측을 해볼 수 있습니다.



● 프로젝트 결과

➢ 결과

성공적으로 ELK 환경을 구축하고 OpenAPI를 사용하여 서울시의 날씨 데이터를 수집했습니다. 이를 토대로 서울시에 노원, 남산, 강동에 평균 일조량, 풍향을 시각화를 해보고 이를 분석해보았습니다.

➢ 추가 논의

FileBeat는 실행 환경의 파일을 주기적으로 서버(Logstash, Elasticsearch etc.)로 올려주는 역활을 합니다. 만약 웹 크롤러(Web Crawler)를 사용하고, 크롤러가 수집하는 데이터를 파일 형태로 저장하며, 상기 파일을 FileBeat가 관리한다면, Elasticsearch는 웹 데이터를 인덱싱 할 수 있을 것입니다.

실제로 시각화한 데이터를 분석하려면 단순 시각화 뿐만 아니라, 시각화 이후 분석방법에 대한 이해가 필요한 것을 느꼈습니다.

➢ 추가 필요 사항

현재의 데이터는 크기가 작아 빅데이터에 수준이 못 미칩니다. 만약 데이터 수집에 양이 거대해 진다면 'Microsoft Azure'와 같은 클라우드 컴퓨팅 자원이 필요할 것입니다.

빅데이터 분석 시, 데이터의 전처리 과정에서 가장 많은 시간이 소요된다고 배웠는데, 이번 프로젝트에서는 전처리를 적용하지 못했습니다. 전처리 방법을 학습 한 뒤, 본 환경에 적용이 필요합니다.