Logstash

①Logstash 설치와 개념, 기본실습

https://www.youtube.com/watch?v=8iXZTS7f_hY

이 유튜브를 보고 똑같이 따라하면 Logstash 설치가 가능하다!

https://www.elastic.co/kr/downloads/logstash

위 사이트는 elasticsearch Logtash 공식사이트며 다운로드 파일와 `logstash -f logstash.conf` 코드가 적혀 있다.

https://www.elastic.co/guide/en/logstash/current/index.html

위 사이트는 elasticsearch Logtash 공식사이트며 버전별 사용되는 코드가 정리되어 있다.

- 왠만하면 logstash파일은 D드라이브 혹은 C드라이브에 압축풀기를 하는 것이 좋다.

  (파일명이 지나치게 길어서 압축풀기가 안되는 경우도 있다.)

- CONF 파일은 반드시 D:₩logstash-7.16.0-windows-x86_64₩logstash-7.16.0₩bin 밑에 있을 필요가 없다. 어느 곳에 있어도 전혀 상관없다.

- CONF 파일명을 "kmh.conf"라고 설정했으면 cmd창에 `logstash -f kmh.conf` 를 입력한다.

- CONF 파일의 코드는 Input/filter/output 으로 구성되어 있다.

<<logstash 설치 후 방법>>

```
Microsoft Windows [Version 10.0.19043.1266]
(c) Microsoft Corporation. All rights reserved.

C:₩Windows₩system32>d:

D:₩>D:₩logstash-7.16.0-windows-x86_64₩logstash-7.16.0₩bin
'D:₩logstash-7.16.0-windows-x86_64₩logstash-7.16.0₩bin'은(는) 내부 또는 외
배치 파일이 아닙니다.

D:₩>cd D:₩logstash-7.16.0-windows-x86_64₩logstash-7.16.0₩bin

D:₩logstash-7.16.0-windows-x86_64₩logstash-7.16.0₩bin>logstash -f hi.conf
```

1. 파일 위치로 이동하기
2. logstash -f hi.conf(=CONF 파일명)
3. 해당되는 값 입력하기 혹은 잘 나오는지 확인하기

logstash.conf

D: > logstash-7.16.0-windows-x86_64 > logstash-7.16.0 > bin > logstash.conf

```
input {
    stdin {}
}

output {
    elasticsearch {
        hosts => ["localhost:9200"]
        index => "indexforlogstash"
        }
}
```

에서 stdin {} 는 cmd창에 직접 값을 집어넣는다는 의미이다.

- **한국어를 넣으면 오류가 생긴다.**

```
14  GET indexforlogstash/_search
```


```
    {
      "_index" : "indexforlogstash",
      "_type" : "_doc",
      "_id" : "H7d3sn0B_FKBDlGTU7IX",
      "_score" : 1.0,
      "_source" : {
        "message" : """I am hungry, maybe
          you feel like that
""",
        "@timestamp" : "2021-12-13T06:25
          :04.150Z",
        "host" : "DESKTOP-G9DVJ1H",
        "@version" : "1"
      }
    },
    {
      "_index" : "indexforlogstash",
      "_type" : "_doc",
      "_id" : "ILeisn0B_FKBDlGTEbIh",
      "_score" : 1.0,
      "_source" : {
        "message" : """Wow, Minhyoung is
          the greastest data engineer!
""",
        "@timestamp" : "2021-12-13T07:11
          :45.301Z",
        "host" : "DESKTOP-G9DVJ1H",
        "@version" : "1"
      }
    }
  ]
 }
}
```


원하는 데로 데이터가 이쁘게 들어갔다.

```
[2021-12-13T16:30:27,223][WARN ][logstash.runner          ] SIGINT received. Shutting down.
[2021-12-13T16:30:27,464][INFO ][logstash.javapipeline    ][main] Pipeline terminated {"pipeline.id"=>"main"}
[2021-12-13T16:30:28,357][INFO ][logstash.runner          ] Logstash shut down.
일괄 작업을 끝내시겠습니까 (Y/N)? y
```

Ctrl+c를 누르고 y를 누르면 작업이 끝이 난다.

https://koocci-dev.tistory.com/19

위 사이트의 실습을 진행해보자.

hi.conf

D: > logstash-7.16.0-windows-x86_64 > logstash-7.16.0 > bin > hi.conf

```
1  input{
2      stdin { }
3  }
4  output{
5      stdout { }
6  }
```

```
Microsoft Windows [Version 10.0.19043.1266]
(c) Microsoft Corporation. All rights reserved.

C:₩Windows₩system32>d:

D:₩>D:₩logstash-7.16.0-windows-x86_64₩logstash-7.16.0₩bin
'D:₩logstash-7.16.0-windows-x86_64₩logstash-7.16.0₩bin'은(는) 내부 또는 외
배치 파일이 아닙니다.

D:₩>cd D:₩logstash-7.16.0-windows-x86_64₩logstash-7.16.0₩bin

D:₩logstash-7.16.0-windows-x86_64₩logstash-7.16.0₩bin>logstash -f hi.conf
```

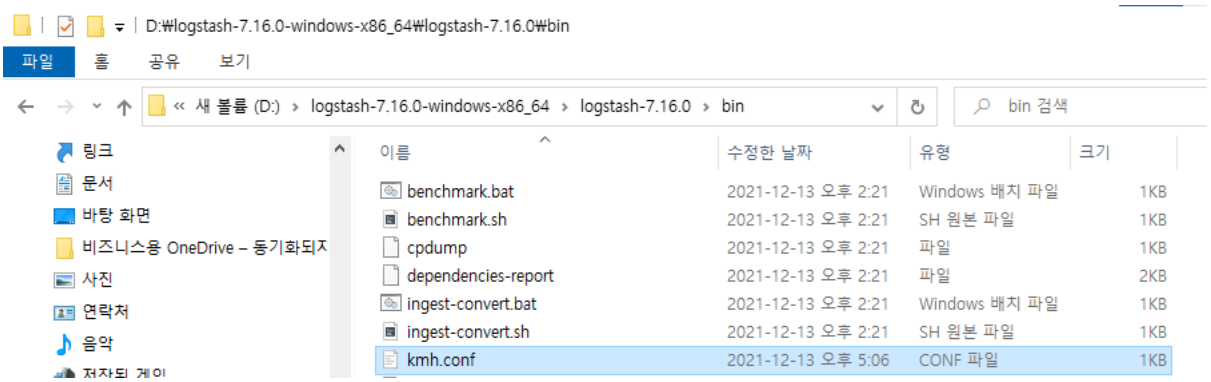
```
helloword
{
      "@version" => "1",
          "host" => "DESKTOP-G9DVJ1H",
    "@timestamp" => 2021-12-14T04:08:12.465Z,
       "message" => "helloword₩r"
}
thank you
{
      "@version" => "1",
          "host" => "DESKTOP-G9DVJ1H",
    "@timestamp" => 2021-12-14T04:08:18.141Z,
       "message" => "thank you₩r"
}
koocci
{
      "@version" => "1",
          "host" => "DESKTOP-G9DVJ1H",
    "@timestamp" => 2021-12-14T04:08:26.883Z,
       "message" => "koocci₩r"
}
```

②logstash 응용실습(CSV 파일 실습)

이번에는 csv파일을 집어넣어보자.

https://peanut159357.tistory.com/27

위의 사이트의 실습을 진행해보자.



kmh.conf

D: > logstash-7.16.0-windows-x86_64 > logstash-7.16.0 > bin > kmh.conf

```
input {
        file {
                path => [ "D:/logstash-7.16.0-windows-x86_64/bicycle2_202006.csv" ]
            start_position => "beginning"
            sincedb_path => "nul"
            codec => plain{ charset => "UTF-8" }
        }
}
filter {
        csv{
                columns => ["대여소 그룹","대여소 명","대여 일자 / 월","대여 건수"]
                separator => ","
        }
}
output {
        elasticsearch {
                hosts => ["localhost:9200"]
                index => "this"
        }
        stdout { }
}
```

에서 stdout { } 는 cmd창에 결과값을 직접 보여줄 거라는 의미이다.

message는 모든 행들의 값이 합친 형태로 나오게 된다. → 보통 필요 없으므로 삭제한다.

관리자: 명령 프롬프트 - logstash -f kmh.conf

```
        "대여소 그룹" => "종로구",
         "대여 건수" => "37",
          "host" => "DESKTOP-G9DVJ1H",
         "대여소 명" => "3418.창신역2번 출구",
    "대여 일자 / 월" => "202006",
     "@version" => "1"
}
{
      "message" => "종로구,3419,국립어린이과학관,202006,145₩r",
    "@timestamp" => 2021-12-13T08:07:16.874Z,
          "path" => "D:/logstash-7.16.0-windows-x86_64/bicycle2_202006.csv",
        "대여소 그룹" => "종로구",
         "대여 건수" => "145",
          "host" => "DESKTOP-G9DVJ1H",
         "대여소 명" => "3419.국립어린이과학관",
    "대여 일자 / 월" => "202006",
     "@version" => "1"
}
{
      "message" => "종로구,342. 대학로 마로니에공원,202006,2557₩r",
    "@timestamp" => 2021-12-13T08:07:16.874Z,
          "path" => "D:/logstash-7.16.0-windows-x86_64/bicycle2_202006.csv",
        "대여소 그룹" => "종로구",
         "대여 건수" => "2557",
          "host" => "DESKTOP-G9DVJ1H",
         "대여소 명" => "342. 대학로 마로니에공원",
    "대여 일자 / 월" => "202006",
     "@version" => "1"
```

결과값을 cmd창에 보여줌을 알 수 있다.

elasticsearch에도 잘 반영되었음을 알 수 있다.

https://koocci-dev.tistory.com/20

위의 사이트의 실습을 진행해보자.

위 사이트는 start_position, sincedb, /dev/null, mutate {convert => [,] } 의미도 잘 알려준다.

- **/dev/null 은 리눅스 버전이고 nul은 윈도우 버전이다.'**
- **mutate {remove_field => ["@version","host","message","path"]} 는 해당되는 행의 값을 없앤다는 의미이다.**
- **git주소로 path를 설정하는 경우 오류가 생겼다. 왠만하면 csv파일을 다운받고 실행하자.**

https://www.elastic.co/guide/en/logstash/current/index.html

위 사이트는 Input, filter, output에 관한 코드들은 위 사이트에 버전별로 잘 정리되어 있다.

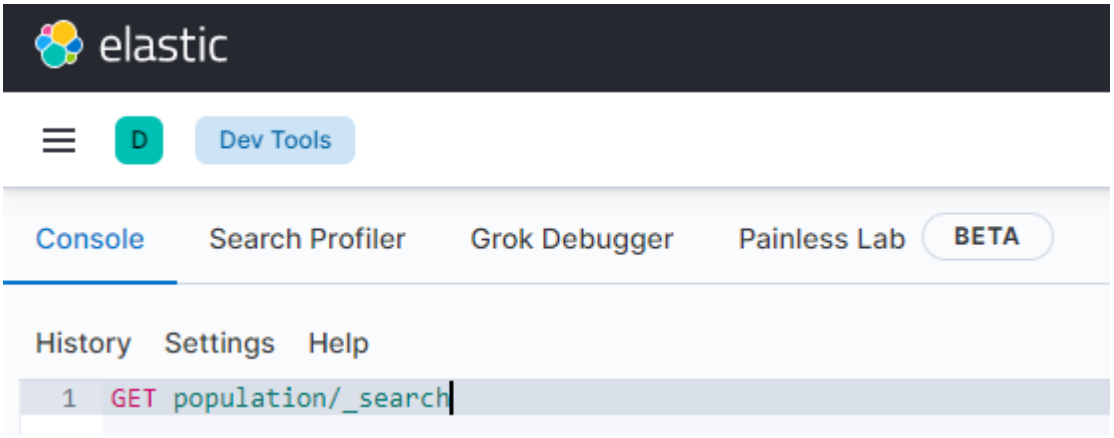
D: > logstash-7.16.0-windows-x86_64 > logstash-7.16.0 > bin > population.conf

```
input {
  file {
    path => "D:/logstash-7.16.0-windows-x86_64/populationbyc
    start_position => "beginning"
    sincedb_path => "nul"
  }
}
filter {
  csv {
      separator => ","
      columns => ["Country","1980","1981","1982","1983","198
  }
  mutate {convert => ["1980", "float"]}
  mutate {convert => ["2008", "float"]}
  mutate {convert => ["2009", "float"]}
  mutate {convert => ["2010", "float"]}
  mutate {remove_field => ["@version","host","message","path"]}

}
output {
  elasticsearch {
      hosts => "localhost:9200"
      index => "population"
  }
  stdout {}
}
```

```
{
         "1980" => 0.09136,
         "2004" => 0.11026,
         "2000" => 0.10235,
         "1999" => 0.10051,
         "1998" => 0.09887,
         "1997" => 0.09744,
         "2001" => 0.10426,
      "Country" => "Tonga",
         "2009" => 0.1209,
         "2006" => 0.11471,
         "1982" => 0.0922,
         "2007" => 0.11694,
         "1987" => 0.094,
         "1991" => 0.09126,
         "1994" => 0.09444,
         "2008" => 0.11901,
         "1981" => 0.09177,
         "1986" => 0.09374,
         "1984" => 0.09288,
         "1983" => 0.09263,
         "2005" => 0.11245,
    "@timestamp" => 2021-12-14T05:17:30.885Z,
         "1990" => 0.09212,
         "1993" => 0.09378,
         "1995" => 0.0952,
         "2002" => 0.10616,
         "1988" => 0.09332,
         "1992" => 0.09214,
         "2010" => 0.12258,
         "1985" => 0.09335,
         "1989" => 0.09278,
         "1996" => 0.0962,
         "2003" => 0.10817
}
```

결과값을 cmd창에 보여줌을 알 수 있다.




```
    },
    {
      "_index" : "population",
      "_type" : "_doc",
      "_id" : "qFVnt30BRtanmOucErdR",
      "_score" : 1.0,
      "_source" : {
        "1996" : 57.39232,
        "1980" : 56.47271,
        "Country" : "Italy",
        "@timestamp" : "2021-12-14T05:25:23.573Z",
        "1985" : 56.7538,
        "1991" : 56.77124,
        "1984" : 56.71946,
        "1999" : 57.63023,
        "2007" : 58.17808,
        "1982" : 56.55763,
        "1983" : 56.65244,
        "2008" : 58.17607,
        "1993" : 57.0511,
        "2009" : 58.15733,
        "2003" : 58.02709,
        "1981" : 56.5242,
        "2002" : 57.95519,
        "2004" : 58.08669,
        "1986" : 56.7566,
        "2005" : 58.13264,
        "1998" : 57.57648,
        "1992" : 56.86492,
        "1990" : 56.76636,
        "1987" : 56.75266,
        "1988" : 56.75707,
        "2010" : 58.12216,
        "1989" : 56.76072,
        "1995" : 57.29941,
        "1994" : 57.20409,
        "2000" : 57.74642,
        "2001" : 57.87254,
        "2006" : 58.16346,
        "1997" : 57.5052
      }
    },
```


elasticsearch에도 잘 반영되었음을 알 수 있다.

③logstash 응용실습(Python 파일 실습)

https://url.kr/l8eica

```
else:

    if htmlObj.status_code == 200:

        bsObj  = BeautifulSoup(htmlObj.text, "html.parser")

        titles = bsObj.select("td.title > div.tit3 > a")

        with open("./nvr_movie_"+ self._cllTime +"_.json", "a", encoding="utf-8") as f:

            for c, t in enumerate(titles):

                d = {
                    "title"  : t.attrs["title"],
                    "rank"   : c+1,
                    "clltime": self._cllTime,
                    "genr"   : self._category["category"][i] }

                f.write(json.dumps(d ,ensure_ascii=False) + "₩n")

            f.close()
```

python코드를 java로 변환함.

```
input {
    stdin { }
}

filter {
    json {
        source => "message"
    }
    mutate {
        remove_field => ["@version", "host", "message", "path"]
    }
}

output {
    stdout { codec => rubydebug }
    elasticsearch {
        hosts => ["http://192.168.42.136:9200"]
        index => nvr_movie
    }
}

logstash 코드
```

변환한 java코드를 logstash에 집어넣음. → java 파일을 집어넣을 때와 같은 모양이다.

④logstash VS Python elasticsearch package(CSV, Python 비교)

√CSV

Logstash

kmh.conf

: > logstash-7.16.0-windows-x86_64 > logstash-7.16.0 > bin > kmh.conf

```
input {
        file {
                path => [ "D:/logstash-7.16.0-windows-x86_64/bicycle2_202006.csv" ]
            start_position => "beginning"
            sincedb_path => "nul"
            codec => plain{ charset => "UTF-8" }
        }
}
filter {
        csv{
                columns => ["대여소 그룹","대여소 명","대여 일자 / 월","대여 건수"]
                separator => ","
        }
}
output {
        elasticsearch {
                hosts => ["localhost:9200"]
                index => "this"
        }
        stdout { }
}
```

Python

```
# elasticsearch에 넣고 싶은 데이터 읽어오기

import pandas as pd
data=pd.read_csv('Report.csv')
data
```

```
from elasticsearch import Elasticsearch
```

```
es = Elasticsearch('http://127.0.0.1:9200', timeout=30, max_retries=10, retry_on_timeout=True)
```

```
for i in range(len(data['자치구'])):
    es.index(index="tableau2",
        doc_type='_doc',
        document={'자치구':data['자치구'][i],
        '세대':data['세대'][i],
        '총인구':data['총인구'][i],
        '남자':data['남자'][i],
        '여자':data['여자'][i]
                })
```

두 방법 다 사용법이 쉽다.

√Python

Logstash

https://url.kr/l8eica

```
  else:

      if htmlObj.status_code == 200:

          bsObj  = BeautifulSoup(htmlObj.text, "html.parser")

          titles = bsObj.select("td.title > div.tit3 > a")

          with open("./nvr_movie_"+ self._cllTime +"_.json", "a", encoding="utf-8") as f:

              for c, t in enumerate(titles):

                  d = {
                      "title"  : t.attrs["title"],
                      "rank"   : c+1,
                      "clltime": self._cllTime,
                      "genr"   : self._category["category"][i] }

                  f.write(json.dumps(d ,ensure_ascii=False) + "₩n")

              f.close()
```

python코드를 java로 변환함.

```
input {
    stdin { }
}

filter {
    json {
        source => "message"
    }
    mutate {
        remove_field => ["@version", "host", "message", "path"]
    }
}

output {
    stdout { codec => rubydebug }
    elasticsearch {
        hosts => ["http://192.168.42.136:9200"]
        index => nvr_movie
    }
}

logstash 코드
```

변환한 java코드를 logstash에 집어넣음. → java 파일을 집어넣을 때와 같은 모양이다.

Python

```
from elasticsearch import Elasticsearch
```

```
es = Elasticsearch('http://127.0.0.1:9200', timeout=30, max_retries=10, retry_on_timeout=True)
```

```
index_name = 'kmh_test4'+"_"+today

for i in range(len(info['url'])):
    es.index(index=index_name,
        doc_type='_doc',
        document={'title':info['title'][i],          # body는
        'titlekeyword':info['titlekeyword'][i],
        'url':info['url'][i],
        'company':info['company'][i],
        'contents':info['contents'][i],
        'date':info['date'][i],
        'image':info['image'][i],
        'keyword':info['keyword'][i],
        'category':info['category'][i]})

es.indices.put_alias(index = index_name, name = 'kmh_test4')
```

python코드를 java로 변환하지 않고 바로 elasticsearch에 집어넣기 때문에 logstash보다 간단하다.