<http://ilhee.tistory.com/category/2016/ELK>

http://geniedev.tistory.com/6

### [[elasticsearch] 엘라스틱서치 시작하기](http://geniedev.tistory.com/6)

Geniedev 2016.05.12 12:18

서비스에 검색 & 추천기능 추가를 위해  
**Elasticsearch** 와 **Solr** 그리고 은전한닢에 관한 내용을 정리한다.

엘라스틱서치와 솔라의 공통점

1. 루씬을 기반으로 한 텍스트 검색 엔진 라이브러리이다.

2. Restful API를 제공한다.

3. Cache를 지원한다.

4. 분산처리에 용이하다.

5. Spring Component를 지원한다.

**spring-data**[spring-data-solr](http://projects.spring.io/spring-data-solr/) 와 [spring-data-elasticsearch](http://projects.spring.io/spring-data-elasticsearch/) 를 참고한다.

엘라스틱서치와 솔라 비교

|  |  |  |
| --- | --- | --- |
|  | Solr | ElasticSearch |
| 도큐먼트 형식 | XML, CSV, JSON | JSON |
| 다중 스키마 문서 | 지원하지 않음 | 지원 |
| 조인 | 지원하지 않음 | Parent\_type/Children\_type |
| 분산 시스템 노드 연결 | 주키퍼 | 젠 디스커버리, 주키퍼 |
| MySQL 불러오기 | dataimportHandler | logstrash |

엘라스틱서치?

엘라스틱서치는 앞서 말했듯, 루씬을 기반으로 한 텍스트 검색 엔진 라이브러리이다. 사전 매핑 없이 JSON 문서 형식으로 입력하면 별도의 이벤트가 없어도 바로 색인을 시작한다. 이렇게 저장된 데이터는 별도의 재시작/갱신 없이도 바로 검색에 사용될 수 있다. 이는 곧 색인 작업이 완료됨과 동시에 검색이 가능하다는걸 의미한다. 이러한 특징들 덕분에 솔라와 비교하여 실시간 검색 엔진 구현에 좀 더 적합하다.

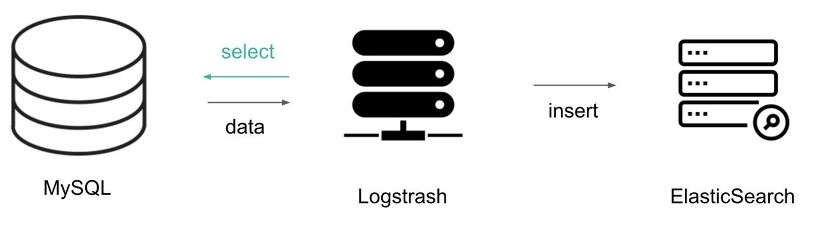
엘라스틱서치와 관계형 데이터베이스

엘라스틱서치를 쉽게 이해하기 위해 사용되는 용어들을 관계형 데이터베이스와 비교하면 쉽게 받아드릴수 있다.

|  |  |
| --- | --- |
| RDB | Elasticsearch |
| DATABASE | INDEX |
| TABLE | TYPE |
| ROW | DOCUMENT |
| COLUMN | FEILD |
| SCHEMA | MAPPING |

엘라스틱서치에서 MySQL 불러오기

1.9버전까지는 reiver라는 친구를 이용하여 MySQL에 데이터를 불러왔었는데, 2.0버전부터 공식적인 지원을 중단하였다. 엘라스틱은 이후 logstash를 이용하여 MySQL의 데이터를 불러오기를 권장하고 있다.



엘라스틱서치 설치

대충알아봤으니 설치해보자.

|  |  |  |
| --- | --- | --- |
| 1 | $ wget https://download.elastic.co/elasticsearch/release/org/elasticsearch/distribution/tar/elasticsearch/2.3.2/elasticsearch-2.3.2.tar.gz | [cs](http://colorscripter.com/info#e) |

압축을 풀고 클러스터를 실행한다.

|  |  |  |
| --- | --- | --- |
| 1 | $ ./bin/elasticsearch --cluster.name my\_cluster\_name --node.name my\_node\_name | [cs](http://colorscripter.com/info#e) |

다음과 같이 실행되는걸 볼 수 있다.

|  |  |  |
| --- | --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | [2016-05-12 02:16:39,444][INFO ][node                     ] [Wicked] version[2.3.2], pid[3898], build[b9e4a6a/2016-04-21T16:03:47Z]  [2016-05-12 02:16:39,445][INFO ][node                     ] [Wicked] initializing ...  [2016-05-12 02:16:39,904][INFO ][plugins                  ] [Wicked] modules [reindex, lang-expression, lang-groovy], plugins [analysis-seunjeon], sites []  [2016-05-12 02:16:39,921][INFO ][env                      ] [Wicked] using [1] data paths, mounts [[/ (/dev/xvda1)]], net usable\_space [4.4gb], net total\_space [7.7gb], spins? [no], types [ext4]  [2016-05-12 02:16:39,921][INFO ][env                      ] [Wicked] heap size [1007.3mb], compressed ordinary object pointers [true]  [2016-05-12 02:16:39,921][WARN ][env                      ] [Wicked] max file descriptors [4096] for elasticsearch process likely too low, consider increasing to at least [65536]  [2016-05-12 02:16:41,595][INFO ][node                     ] [Wicked] initialized  [2016-05-12 02:16:41,599][INFO ][node                     ] [Wicked] starting ...  [2016-05-12 02:16:41,654][INFO ][transport                ] [Wicked] publish\_address {172.000.000.000:9300}, bound\_addresses {[::]:9300}  [2016-05-12 02:16:41,662][INFO ][discovery                ] [Wicked] elasticsearch/jdsJR3LSTkKcCRc8YE4rlQ  [2016-05-12 02:16:44,769][INFO ][cluster.service          ] [Wicked] new\_master {Wicked}{jdsJR3LSTkKcCRc8YE4rlQ}{172.31.6.226}{172.31.6.226:9300}, reason: zen-disco-join(elected\_as\_master, [0] joins received)  [2016-05-12 02:16:44,793][INFO ][http                     ] [Wicked] publish\_address {172.000.000.000:9200}, bound\_addresses {[::]:9200}  [2016-05-12 02:16:44,793][INFO ][node                     ] [Wicked] started    [*Colored by Color Scripter*](http://colorscripter.com/info#e) | [cs](http://colorscripter.com/info#e) |

기본적으로 ::1:9300 으로 바운딩 되어 있는데 config/elasticsearch.yml 파일에서

|  |  |  |
| --- | --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | # ---------------------------------- Network -----------------------------------  #  # Set the bind address to a specific IP (IPv4 or IPv6):  #   network.host: 0.0.0.0  #  # Set a custom port for HTTP:  #  # http.port: 9200    [*Colored by Color Scripter*](http://colorscripter.com/info#e) | [cs](http://colorscripter.com/info#e) |

이부분을 수정해주면 된다.

로그스태시 설치

MySQL을 import 하기 위해서 logstash를 설치하자.

|  |  |  |
| --- | --- | --- |
| 1 | $ wget https://download.elastic.co/logstash/logstash/logstash-2.1.1.tar.gz | [cs](http://colorscripter.com/info#e) |

기본설정파일을 작성하고

|  |  |  |
| --- | --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | /\* logstash.conf \*/  input {      stdin {}  }    output {      elasticsearch {          hosts => ["localhost:9200"]      }      stdout {          codec => rubydebug      }  }  [*Colored by Color Scripter*](http://colorscripter.com/info#e) | [cs](http://colorscripter.com/info#e) |

적용한다.

|  |  |  |
| --- | --- | --- |
| 1 | $ bin/logstash -f logstash.conf --configtest | [cs](http://colorscripter.com/info#e) |

Jdbc plugin을 설치한다.

|  |  |  |
| --- | --- | --- |
| 1  2  3  4 | $ bin/plugin list jdbc$ bin/plugin install logstash-input-jdbc  $ wget http://xbib.org/repository/org/xbib/elasticsearch/importer/elasticsearch-jdbc/2.1.1.2/elasticsearch-jdbc-2.1.1.2-dist.zip  $ sudo apt-get install unzip  $ cp elasticsearch-jdbc-2.1.1.2/lib/mysql-connector-java-5.1.33.jar lib/  [*Colored by Color Scripter*](http://colorscripter.com/info#e) | [cs](http://colorscripter.com/info#e) |

Jdbc plugin을 위한 설정파일을 작성하고

|  |  |  |
| --- | --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19 | /\* logstash.conf \*/  input {      jdbc {          jdbc\_driver\_library => "lib/mysql-connector-java-5.1.33.jar"          jdbc\_driver\_class => "com.mysql.jdbc.Driver"          jdbc\_connection\_string => "jdbc:mysql://localhost:3306/[db-name]"          jdbc\_user => "[db-user]"          jdbc\_password => "[password]"          statement => "SELECT \* FROM [tbl-name]"          schedule => "\* \* \* \* \*"      }  }  output {      elasticsearch {          hosts => ["localhost:9200"]      }    stdout {          codec => rubydebug      }  }  [*Colored by Color Scripter*](http://colorscripter.com/info#e) | [cs](http://colorscripter.com/info#e) |

적용하면 끝

|  |  |  |
| --- | --- | --- |
| 1  2 | $ bin/logstash -f logstash-jdbc.conf --configtest  $ bin/logstash -f logstash-jdbc.conf | [cs](http://colorscripter.com/info#e) |

은전한닢 설치

한국에서 서비스할꺼고, 검색어는 한국어일테니 형태소로 나누어 검색이 가능하게 해야한다.  
훌륭한 오픈소스 한국어 형태소 분석기인 은전한닢을 적용하자.

|  |  |  |
| --- | --- | --- |
| 1 | $ ./bin/plugin install org.bitbucket.eunjeon/elasticsearch-analysis-seunjeon/2.3.2.0 | [cs](http://colorscripter.com/info#e) |

**은전한닢**<https://bitbucket.org/eunjeon/seunjeon/src/6e8a067fb9a12bcdcdd7f858fd84714c94835f04/elasticsearch/>

인덱스 생성

설치가 모두 끝났으니 간단히 사용해 보자. 우선 데이터베이스에 해당하는 인덱스를 생성한다.

|  |  |  |
| --- | --- | --- |
| 1 | $ curl -X POST http://ip:9200/test | [cs](http://colorscripter.com/info#e) |

|  |  |  |
| --- | --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | {      "mappings" : {          "users" : {              "properties" : {                  "index" : { "type" : "integer" },                  "email" : { "type" : "string" },                  "nickName" : { "type" : "string"},                  "tag" : { "type" : "string"}              }          }      }  }  [*Colored by Color Scripter*](http://colorscripter.com/info#e) | [cs](http://colorscripter.com/info#e) |

이번엔 은전한닢이 분석기로 적용된 인덱스를 생성한다.

|  |  |  |
| --- | --- | --- |
| 1 | $ curl -X POST http://ip:9200/enjun | [cs](http://colorscripter.com/info#e) |

|  |  |  |
| --- | --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28 | {      "settings" : {          "analysis" : {                                                   // 은전한닢 형태소 분석기 적용              "analyzer" : {                  "korean\_analyzer" : {                      "type":"custom",                      "tokenizer":"seunjeon\_default\_tokenizer"                  }              },              "tokenizer": {                  "seunjeon\_default\_tokenizer": {                      "type": "seunjeon\_tokenizer"                  }              }          }      },      "mappings" : {          "users" : {              "properties" : {                  "index" : { "type" : "integer" },                  "email" : { "type" : "string" },                  "nickName" : { "type" : "string", "analyzer" : "korean\_analyzer" },                  "tag" : { "type" : "string", "analyzer" : "korean\_analyzer"                  }              }          }      }  }  [*Colored by Color Scripter*](http://colorscripter.com/info#e) | [cs](http://colorscripter.com/info#e) |

도큐먼트 생성

두 인덱스 모두 도큐멘트를 생성한다.

|  |  |  |
| --- | --- | --- |
| 1 | $ curl -X POST http://ip:9200/test/uesrs/1 | [cs](http://colorscripter.com/info#e) |

|  |  |  |
| --- | --- | --- |
| 1 | $ curl -X POST http://ip:9200/test/enjun/1 | [cs](http://colorscripter.com/info#e) |

|  |  |  |
| --- | --- | --- |
| 1  2  3  4  5 | {      "email":"test@gmail.com",      "nickName":"지니",      "tag":"보도자료"  } | [cs](http://colorscripter.com/info#e) |

도큐먼트 검색

태그가 '보도v자료' 와 일치하는 사용자를 찾을 것이다. 우선 형태소분석기가 적용되지 않은 인덱스부터

|  |  |  |
| --- | --- | --- |
| 1 | $ curl -X GET http://ip:9200/test/uesrs/\_search?q=tag:'보도 자료' | [cs](http://colorscripter.com/info#e) |

|  |  |  |
| --- | --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | {      "took": 8,      "timed\_out": false,      "\_shards": {          "total": 5,          "successful": 5,          "failed": 0      },      "hits": {          "total": 0,          "max\_score": null,          "hits": []                 // 인식하지 못함      }  }  [*Colored by Color Scripter*](http://colorscripter.com/info#e) | [cs](http://colorscripter.com/info#e) |

'보도자료'를 하나의 키워드로 인식하여 검색결과가 하나도 없는것을 볼 수 있다.

이번엔 형태소 분석기가 적용된 인덱스에서 검색한다.

|  |  |  |
| --- | --- | --- |
| 1 | $ curl -X GET http://ip:9200/enjun/uesrs/\_search?q=tag:'보도 자료' | [cs](http://colorscripter.com/info#e) |

|  |  |  |
| --- | --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23 | {      "took": 8,      "timed\_out": false,      "\_shards": {          "total": 5,          "successful": 5,          "failed": 0  },      "hits": {          "total": 1,          "max\_score": 0.028130025,          "hits": [{              "\_index": "enjun",              "\_type": "users",              "\_id": "1",              "\_score": 0.028130025,              "\_source": {                  "email": "test@gmail.com",                  "nickName": "지니",                  "tag": "보도자료"                       // ‘보도’ + ‘자료’ 로 인식              }          }]      }  }  [*Colored by Color Scripter*](http://colorscripter.com/info#e) | [cs](http://colorscripter.com/info#e) |

참고자료

ElasticSearch Search API Documents

<https://www.elastic.co/guide/en/elasticsearch/reference/current/search-search.html>

ElasticSearch - MySQL Import

<http://digndig.kr/elasticsearch/756/>

<https://github.com/jprante/elasticsearch-jdbc>

ElasticSearch - 개요

<http://www.slideshare.net/seunghyuneom/elastic-search-52724188>