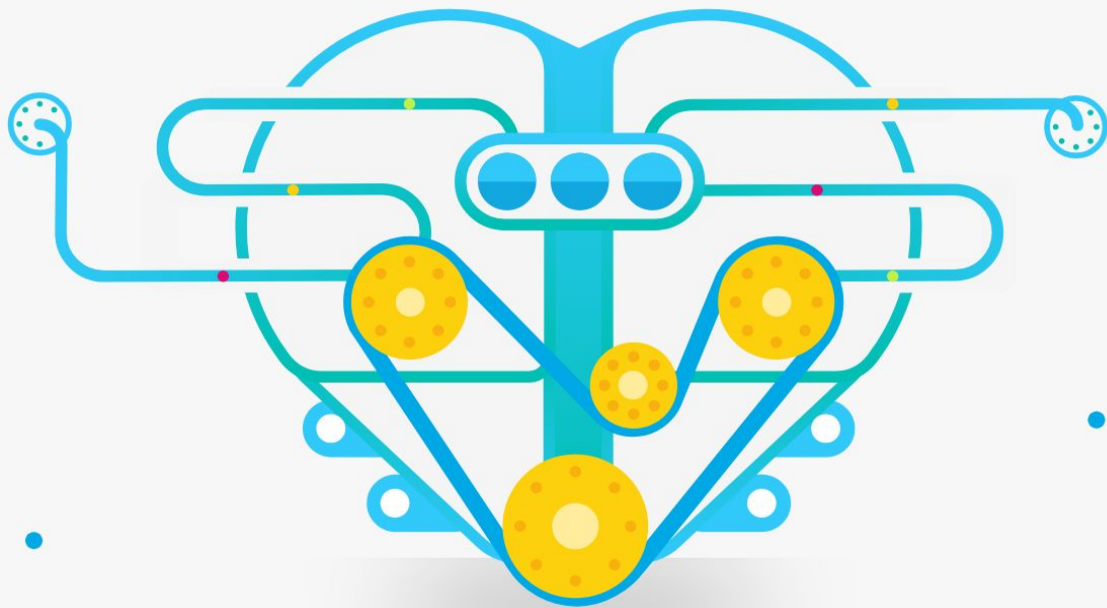


엘라스틱서치 소개 및 개요



간단 소개

개발자	Shay Banon
최근 버전	6.2.2 / 2018년 2월 20일(32일 전) ^[1]
개발 상태	개발 중
프로그래밍 언어	자바
운영 체제	크로스 플랫폼
종류	<u>검색</u> 및 <u>색인</u>
라이선스	아파치 라이선스 2.0
웹사이트	www.elastic.co/products/elasticsearch



주요 특징

아파치 루씬 기반

루씬의 특징

자바 언어로 개발
사용자 위치 정보 이용 가능
다국어 검색 지원
자동 완성 지원
미리 보기 지원
철자 수정 기능 지원

분산 시스템

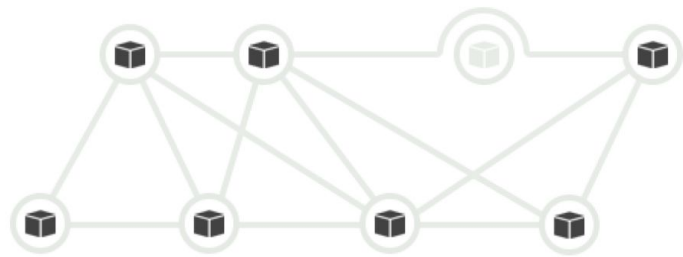
- 엘라스틱 서치는 여러 개의 노드로 구성되는 분산 시스템
- 노드는 데이터를 색인하고 검색을 수행하는 단위 프로세스
- 기존 노드에 새 노드를 실행하여 연결하는 것만으로 확장 가능
- 데이터는 각 노드에 분산 저장
- 복사본을 유지하여 각종 충돌로부터 노드 데이터 보호
- DISCOVERY를 내장하여 별도의 분산 시스템 관리자 불필요

높은 가용성 (HIGH AVAILABILITY)

- 엘라스틱 서치는 **하나 이상의 노드로 구성**
- 각 노드는 1개 이상의 데이터 원본과 복사본을 서로 다른 위치에 나누어 저장
- 노드가 종료되거나 실행에 실패할 경우 다른 노드로 데이터 이동



항상 일정한 데이터 복사본의 개수를 유지하여 **높은 가용성과 안정성 보장**

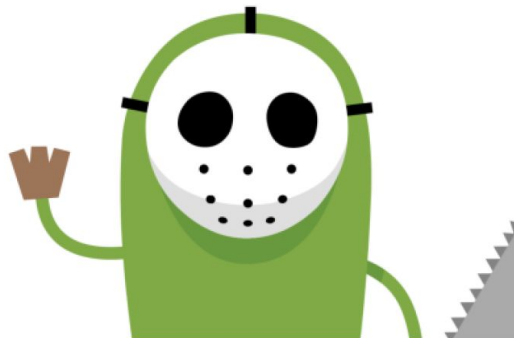


멀티 테넌시 (MULTY TENANCY)

- 데이터는 여러 개로 분리된 인덱스들에 그룹으로 저장 (인덱스 = RDBMS의 데이터베이스에 대응)
- 서로 다른 인덱스의 데이터를 **하나의 질의로 검색하여 하나의 출력으로 도출 가능**

JSON DOCUMENT / RESTFUL API

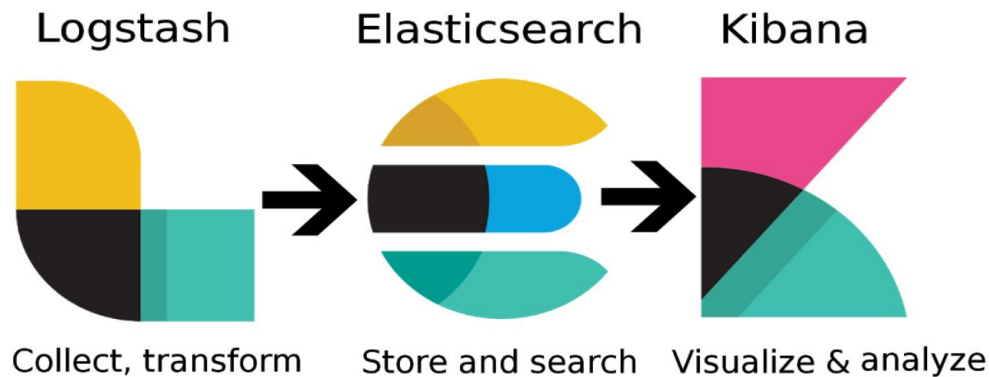
- 기본적으로 모든 필드를 색인 후 JSON 구조로 저장
- JSON 구조로 인해 모든 레벨의 필드에 접근이 쉽고, 빠른 속도로 검색 가능
- 사전 매핑 없이 JSON 문서 형식으로 데이터를 입력하면 바로 색인 작업 수행
- REST 자원은 색인된 데이터 및 질의, 검색되어 JSON 형식으로 출력된 문서를 의미
- JSON 문서를 URI로 명시, 이 문서를 처리하기 위해 HTTP METHOD 이용



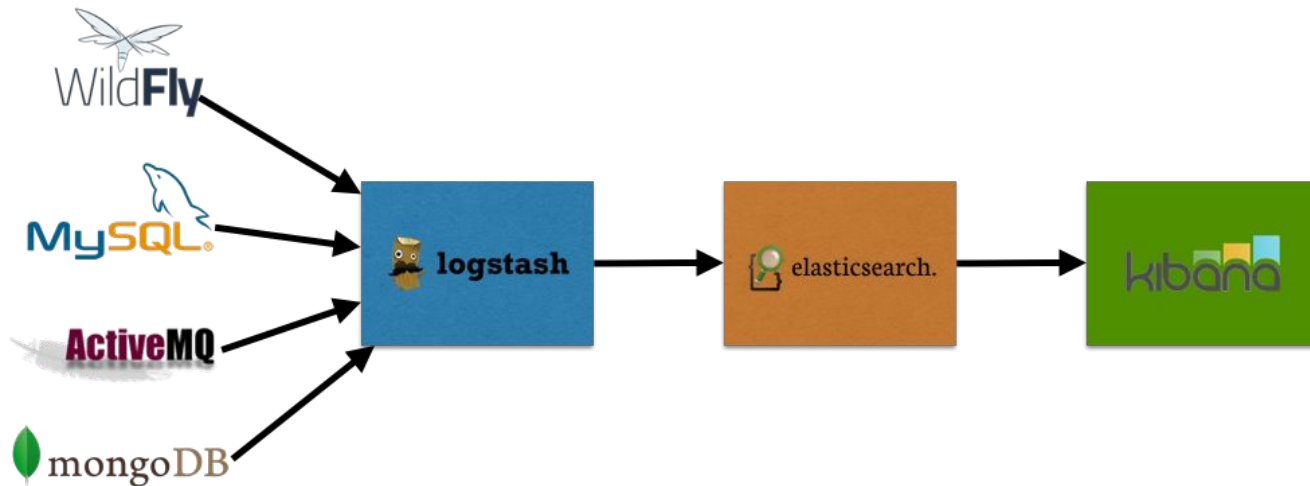


실시간 분석 (REAL TIME)

- 저장된 데이터는 검색에 사용되기 위해 별도의 재시작 / 갱신이 불필요
- 색인 작업이 완료됨과 동시에 바로 검색 가능
- 실시간 분석 / 검색은 데이터 증가량에 구애 받지 않음



ELASTIC STACK 의 심장





Kibana



Elasticsearch



Beats



Logstash



X-Pack



Security



Alerting



Machine Learning



Monitoring



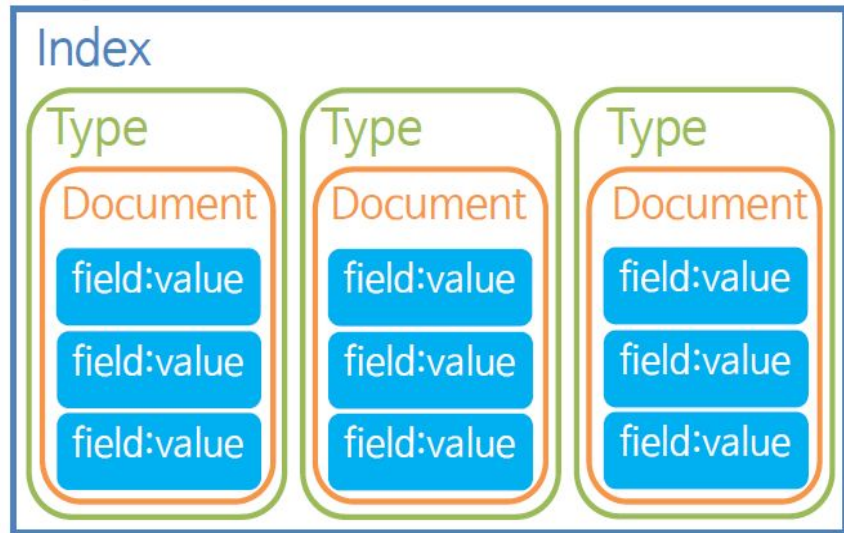
Reporting



Graph

데이터 & 시스템 구조

[Logical 구성]



인덱스 (INDEX)

타입 (TYPE)



타입 (TYPE)



```

1 {
2   "took": 30,
3   "timed_out": false,
4   "_shards": {
5     "total": 5,
6     "successful": 5,
7     "failed": 0
8   },
9   "hits": {
10    "total": 4,
11    "max_score": 0.22821909,
12    "hits": [
13      {
14        "_index": "books",
15        "_type": "book",
16        "_id": "AU8Xa3uw8agau8xyir3f",
17        "_score": 0.22448172,
18        "_source": {
19          "title": "Around the World in Eighty Days",
20          "author": "Jules Verne",
21          "category": "adventure novel",
22          "written": "1873-07-01T10:30:00",
23          "pages": 189,
24          "sell": 27200000,
25          "plot": "Fogg and Passepartout reach Suez in time. While
detective named Fix, who has been dispatched from London in search
robber, Fix mistakes Fogg for the criminal. Since he cannot secure
travellers to Bombay. Fix becomes acquainted with Passepartout with
large reward if he gets them to Bombay early. They dock two days at
26        }
27      },
28    ],
29    "_index": "books",
30    "_type": "book",
31    "id": "AU8Xa3uw8aqau8xvir3d".

```

인덱스 (INDEX)

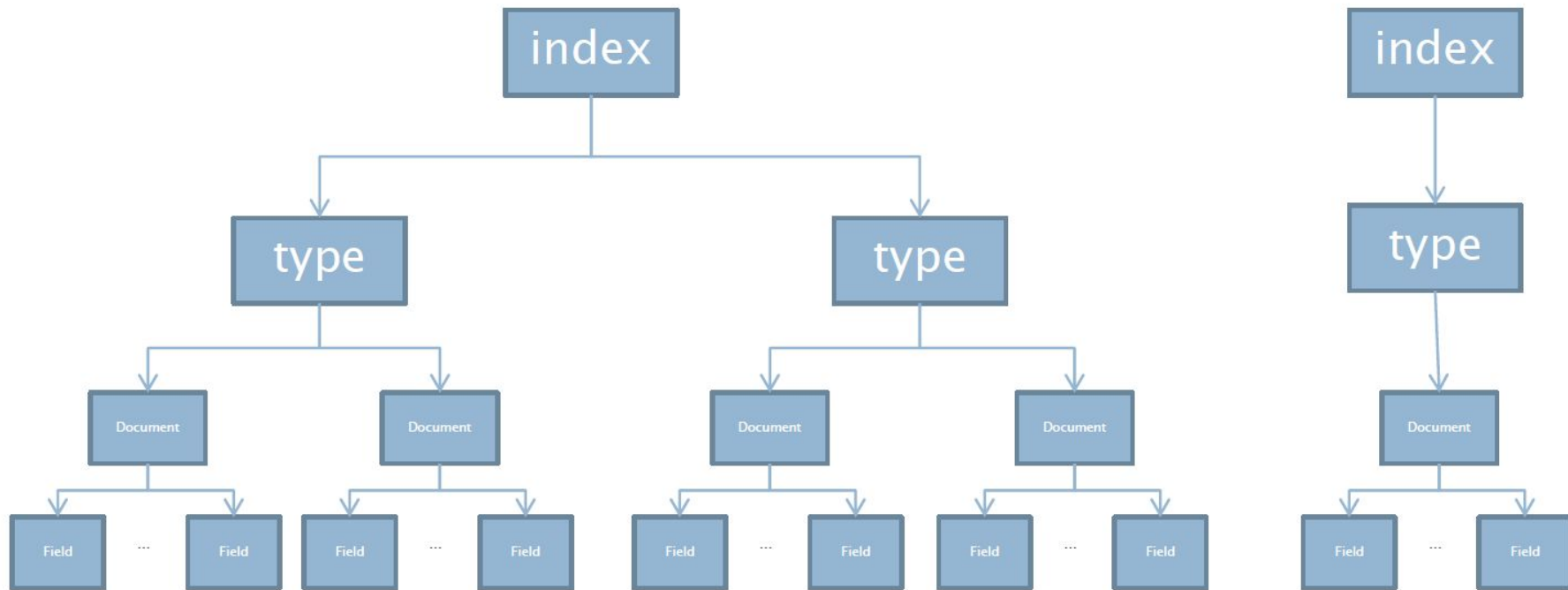
타입 (TYPE)

도큐먼트(DOCUMENT)

```

{
  "name": "Tony",
  "sex": "male"
}

```





관계형 DB	엘라스틱서치
데이터베이스 (DATABASE)	인덱스 (INDEX)
테이블 (TABLE)	타입 (TYPE)
열 (ROW)	도큐먼트 (DOCUMENT)
행 (COLUMN)	필드 (FIELD)
스키마 (SCHEMA)	매핑 (MAPPING)

elasticsearch

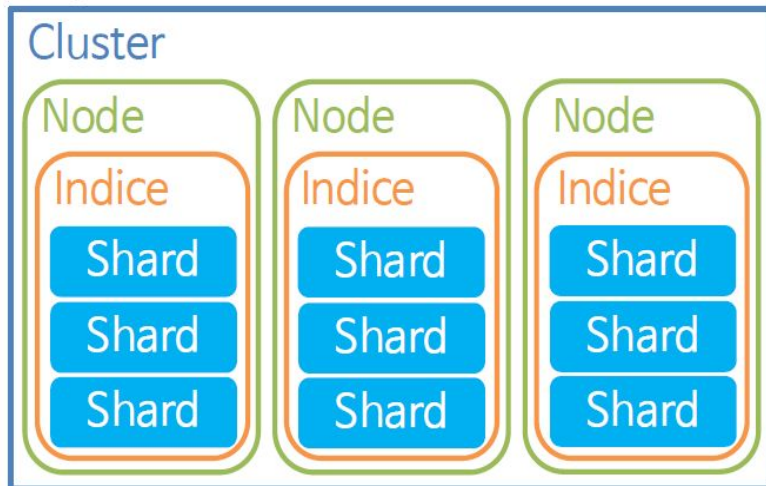


Elasticsearch Client

NGINX Plus

Scalable, Highly-available
Elasticsearch Cluster

[Physical 구성]



Master node

Data node

Search load balancer node

Client node

Master

`node.master: true`

Data

`node.data: true`

Search LB

`node.master: false`
`node.data: false`

Client

`node.client: true`

```
$ bin/elasticsearch --node.name=Node1 --node.master=true --node.data=false  
$ bin/elasticsearch --node.name=Node2 --node.master=false --node.data=true  
$ bin/elasticsearch --node.name=Node3 --node.master=false --node.data=true  
$ bin/elasticsearch --node.name=Node4 --node.master=false --node.data=true
```

books

size: 3.63ki (7.08ki)

docs: 1 (1)

Info ▾

Actions ▾



Node1

Info ▾

Actions ▾



Node2

Info ▾

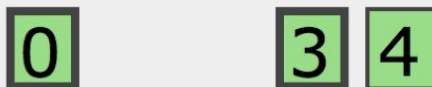
Actions ▾



Node3

Info ▾

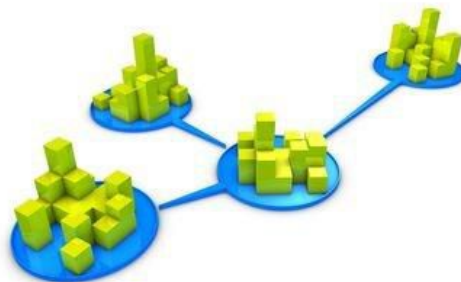
Actions ▾



Node4

Info ▾

Actions ▾



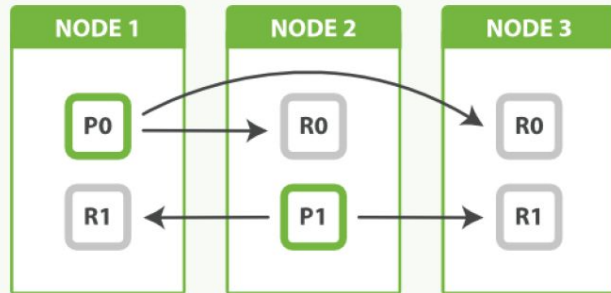
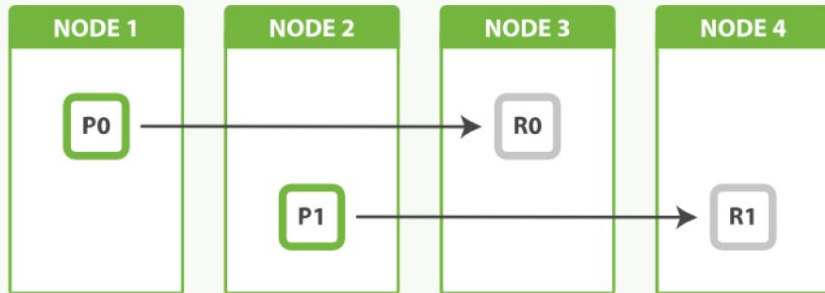
샤드와 복사본 (Shard & Replica)

- 샤드는 데이터 검색을 위해 구분되는 최소 단위
- 색인된 데이터는 여러 개의 샤드로 분할돼 저장
- 기본적으로 인덱스당 5개의 샤드와 5개의 복사본으로 분리
- 개수 설정을 제외하면 사용자가 직접 샤드에 접근하는 경우는 없음
- 데이터가 색인돼 저장되는 공간을 최초 샤드(Primary Shard)라 함
- 최초 샤드에 데이터가 색인되면 동일한 수만큼 복사본을 생성
- 최초 샤드가 유실되는 경우 복사본을 최초 샤드로 승격
- 최초 샤드와 복사본은 동시 검색 대상이 되어 성능 향상에 이점이 있음
- 최초 샤드와 복사본은 서로 다른 노드에 저장

복사본(Replica) 생성 예

```
POST /my_index/_settings
{
  "number_of_replicas": 1
}
```

```
POST /my_index/_settings
{
  "number_of_replicas": 2
}
```



<http://www.elasticsearch.org/guide/en/elasticsearch/guide/current/replica-shards>

설치 & 실행

Installation example with tar

For simplicity, let's use the [tar](#) file.

Let's download the Elasticsearch 6.2.3 tar as follows:

```
curl -L -O https://artifacts.elastic.co/downloads/elasticsearch/elas
```

Then extract it as follows:

```
tar -xvf elasticsearch-6.2.3.tar.gz
```

It will then create a bunch of files and folders in your current directory. We then go into the bin directory as follows:

```
cd elasticsearch-6.2.3/bin
```

And now we are ready to start our node and single cluster:

```
./elasticsearch
```

```
java -version  
echo $JAVA_HOME
```

환경 설정 (elasticsearch.yml)

- 엘라스틱 서치의 대부분의 환경설정을 가지고 있음
- 처음 설치 시 모든 옵션은 주석처리 되어 기본값으로 실행
- 설정 문법은 YAML 문법을 사용하며 콜론 뒤에 공백이 없으면 오류 발생

자주 사용되는 설정

설정	내용
cluster.name	클러스터 이름을 설정 (기본값은 “elasticsearch”)
node.name	노드의 이름을 설정 (기본값은 임의의 이름이 부여)
bootstrap.mlockall	사용 메모리 고정 (JVM이 메모리를 다른 자바프로그램으로 돌리는 것을 방지)
index.number_of_shard	샤드 개수 지정 (기본값은 5)
index.number_of_replicas	복사본 개수 지정 (기본값은 1)

HTTP 네트워크 설정

설정	내용
network.bind_host	서버 내부 IP 주소 (방화벽이나, 공유기 등을 사용 시)
network.publish_host	공개 IP 주소
network.host	내부 IP와 공개 IP가 동일할 경우 사용
transport.tcp.port	다른 노드와 바인딩하여 통신 하기 위해 사용하는 포트 (기본값 9300)
transport.tcp.compress	통신 내용의 압축 여부
http.port	REST API 서비스 포트 (기본값 9200)
http.max_content_length	설정된 용량을 초과한 데이터는 전송하지 않음
http.enabled	REST API 서비스 제공 여부

문서 인덱싱 & 검색

색인 구조 정의(Mapping)

```
$ curl -XPUT localhost:9200/books -d '{
  "mappings" : {
    "book" : {
      "properties" : {
        "author" : {
          "type" : "string",
          "index" : "not_analyzed"
        }
      }
    }
  }
}'
```

문서 색인 (PUT)

```
$ curl -XPUT localhost:9200/books/book/1 -d '  
{ "title" : "introduction of elasticsearch", "author": "dennis park", "publish_date" :  
"2017-03-01" }  
,  
  
$ curl -XPUT localhost:9200/books/book/2 -d '  
{ "title": "development application of machine learning", "author": "james lee",  
"publish_date" : "2015-02-01" }  
,  
  
$ curl -XPUT localhost:9200/books/book/3 -d '  
{ "title" : "easy pyhon guide", "author": "tommy kim", "publish_date" :  
"2018-05-25" }  
,
```

문서 색인 (_BULK)

```
$ curl -XPOST localhost:9200/_bulk -d '  
{ "index" : { "_index" : "books", "_type" : "book", "_id": "1" } }  
{ "title" : "introduction of elasticsearch", "author" : "dennis park", "publish_date" :  
"2017-03-01" }  
{ "index" : { "_index" : "books", "_type" : "book", "_id": "2" } }  
{ "title" : "development application of machine learning", "author" : "james lee",  
"publish_date" : "2015-02-01" }  
{ "index" : { "_index" : "books", "_type" : "book", "_id": "3" } }  
{ "title" : "easy pyhon guide", "author": "tommy kim", "publish_date" :  
"2018-05-25" }  
{ "index" : { "_index" : "books", "_type" : "book", "_id": "4" } }  
{ "title" : "structure of elasticsearch", "author" : "dennis choi", "publish_date" :  
"2017-04-01" }  
,
```

URI 방식 검색

- `_search` API를 사용
- 질의는 `q` 매개변수의 값으로 입력

```
$ curl -XGET 'localhost:9200/books/_search?q=dennis&pretty=true'
```

REQUEST BODY 검색

- 검색 조건을 **JSON 데이터 형식의 질의**로 입력
- URI 검색보다 복잡한 형식으로 검색 가능
- 엘라스틱서치의 **질의 언어(QueryDSL)** 사용

```
$ curl -XPOST 'localhost:9200/books/_search?pretty=true' -d '[QueryDSL]'
```

데모