

기본 개념들

노드 Node

데이터를 저장하고 클러스터에 참여하여 인덱싱 및 검색을 수행

인덱스 Index

비슷한 특성을 가진 문서 간의 집합

타입 Type

인덱스 안에 1개 이상의 타입을 정의 할 수 있음

문서 Document

인덱싱 된 정보의 기본단위

Shards

인덱스를 나눠서 저장하는 것

데이터 분량을 수평적으로 분할하거나 확장가능

Shard 별로 분산/병렬 처리해서 성능 향상

Replica

데이터 손실에 대한 고가용성

검색을 복제본에서 병렬처리 함으로 성능 향상

명령어 [REST Verb] /[Index]/[Type]/[ID]

Elasticsearch 상태 확인

(get) localhost:9200/_cat/health?v&pretty

Node 정보 확인

(get) localhost:9200/_cat/nodes?v&pretty

Index 목록 확인

(get) localhost:9200/_cat/indices?v&pretty

customer Index 생성과 확인

(put) localhost:9200/customer?pretty&pretty

(get) localhost:9200/_cat/indexes?v&pretty

health	status	index	uuid	pri	rep	docs.count	docs.deleted	store.size	pri.store.size
yellow	open	customer	uE5fkrwQ5zCFhgXDWhk8zA	5	1	0	0	810b	810b

Customer index에 문서 넣기

형식 (put) localhost:port/indexName/type/id?pretty&pretty

(put) localhost:9200/customer/external/1?pretty&pretty

```
{
  "name" : "Hong"
}
```

<- body

```
{
  "_index": "customer",
  "_type": "external",
  "_id": "1",
  "_version": 1,
  "result": "created",
  "_shards": {
    "total": 2,
    "successful": 1,
    "failed": 0
  },
  "created": true
}
```

넣은 문서 확인

(get) localhost:9200/customer/external/1?pretty&pretty

_source는 검색된 문서에 대한 JSON

```
{
  "_index": "customer",
  "_type": "external",
  "_id": "1",
  "_version": 1,
  "found": true,
  "_source": {
    "name": "Hong"
  }
}
```

Index 삭제

(delete) localhost:9200/customer?pretty&pretty

삭제 후 확인

(get) localhost:9200/_cat/indices?v&pretty

문서 교체

put으로 문서를 인덱스에 넣고 같은 id로 또 넣으면 문서 교체됨

문서 수정 (1)

(post) localhost:9200/customer/external/1/_update?pretty&pretty

```
{
  "doc" : { "name" : "Gi-Wook",
            "age" : 25 }
}
```

```
{
  "_index": "customer",
  "_type": "external",
  "_id": "1",
  "_version": 2,
  "result": "updated",
  "_shards": {
    "total": 2,
    "successful": 1,
    "failed": 0
  }
}
```

확인

(get) localhost:9200/customer/external/1?pretty&pretty

```
{
  "_index": "customer",
  "_type": "external",
  "_id": "1",
  "_version": 2,
  "found": true,
  "_source": {
    "name": "Gi-Wook",
    "age": 25
  }
}
```

문서 수정 (2)

(post) localhost:9200/customer/external/1/_update?pretty&pretty

```
{  
  "script" : "ctx._source.age += 5"  
}
```

```
{  
  "_index": "customer",  
  "_type": "external",  
  "_id": "1",  
  "_version": 3,  
  "result": "updated",  
  "_shards": {  
    "total": 2,  
    "successful": 1,  
    "failed": 0  
  }  
}  
  
{  
  "_index": "customer",  
  "_type": "external",  
  "_id": "1",  
  "_version": 3,  
  "found": true,  
  "_source": {  
    "name": "Gi-Wook",  
    "age": 30  
  }  
}
```

문서 삭제

(delete) localhost:9200/customer/external/2?pretty&pretty

```
{  
  "found": false,  
  "_index": "customer",  
  "_type": "external",  
  "_id": "2",  
  "_version": 1,  
  "result": "not_found",  
  "_shards": {  
    "total": 2,  
    "successful": 1,  
    "failed": 0  
  }  
}
```

배치 처리

Customer index에 2개 문서 인덱싱

(post)

localhost:9200/customer/external/_bulk?pretty&pretty

```
{  "index" : {"_id" : "1"}}
{  "name" : "Gi-wook"}
{  "index" : {"_id" : "2"}}
{  "name" : "kim"}
```

```
{
  "took": 15,
  "errors": false,
  "items": [
    {
      "index": {
        "_index": "customer",
        "_type": "external",
        "_id": "1",
        "_version": 6,
        "result": "updated",
        "_shards": {
          "total": 2,
          "successful": 1,
          "failed": 0
        },
        "created": false,
        "status": 200
      }
    }
  ]
}
```

한번에 여러 명령어 쓰기

(post) localhost:9200/customer/external/_bulk?pretty&pretty

```
{ "update" : {"_id" : "1"}}
{ "doc" : {"name" : "honghong"}}
{ "delete" : {"_id" : "2"}}
```

```
{
  "took": 14,
  "errors": false,
  "items": [
    {
      "update": {
        "_index": "customer",
        "_type": "external",
        "_id": "1",
        "_version": 7,
        "result": "updated",
        "_shards": {
          "total": 2,
          "successful": 1,
          "failed": 0
        },
        "status": 200
      }
    }
  ]
}
```

검색 API

1. url로 검색

(get)

localhost:9200/bank/_search?q=*&sort=account_number:asc&pretty

q=*은 index의 모든 문서 검색

bank index의 모든 문서에서 account_number를 오름차순으로 검색

- `took` - Elasticsearch가 검색을 수행하는데 걸린 밀리초 단위의 시간입니다.
- `timed_out` - 검색이 시간초과 발생하였는지 여부
- `_shards` - 검색에 사용된 파편의 개수, 검색이 성공 또는 실패한 파편의 개수
- `hits` - 검색결과
- `hits.total` - 검색조건에 일치하는 문서의 개수
- `hits.hits` - 실제 검색된 문서 목록 (기본값은 최초 10개 문서)
- `sort` - 결과의 정렬 속성(없을 경우 score 항목을 사용)

```
{
  "took": 38,
  "timed_out": false,
  "_shards": {
    "total": 5,
    "successful": 5,
    "failed": 0
  },
  "hits": {
    "total": 999,
    "max_score": null,
    "hits": [
      {
        "_index": "bank",
        "_type": "account",
        "_id": "0",
        "_score": null,
        "_source": {
          "account_number": 0,
          "balance": 16623,
          "firstname": "Bradshaw",
          "lastname": "Mckenzie",
          "age": 29,
          "gender": "F",
          "address": "244 Columbus Place",
          "employer": "Euron",
          "email": "bradshawmckenzie@euron.com",
          "city": "Hobucken",
          "state": "CO"
        }
      },
      {
        "sort": [
          0
        ]
      }
    ]
  }
}
```

2. request body로 검색

(post) localhost:9200/bank/_search?pretty

```
{
  "query" : {"match_all" : {}},
  "sort" : [
    {"account_number" : "asc"}
  ]
}
```

Elasticsearch의 경우에는 검색에 대한 요청을 받고 결과를 반환하게 되면 서버에 별도의 자원이나 커서를 보관하지 않습니다. SQL과 같은 다른 플랫폼에서는 검색을 요청하면 쿼리에 대한 부분적인 결과를 수신받고 다시 요청하면 서버 측에서 보관한 검색에 대한 커서를 이어서 진행하여 페이징과 같은 기능을 수행하는 것과는 많이 다릅니다.

쿼리 언어

(post) localhost:9200/bank/_search?pretty

```
{
  "query" : { "match_all" : {} }
}
```

```
    "_source": {
      "account_number": 248,
      "balance": 49989,
      "firstname": "West",
      "lastname": "England",
      "age": 36,
      "gender": "M",
      "address": "717 Hendrickson Place",
      "employer": "Obliq",
      "email": "westengland@obliq.com",
      "city": "Maury",
      "state": "WA"
    },
```

query : 검색할 쿼리문 지칭

match_all : bank 인덱스의 전체문서 조회

size : 결과 목록 개수

from : 시작 위치

특정 인덱스의 모든 문서에서 balance 기준으로 내림차순

```
{
  "query" : { "match_all" : {} },
  "sort" : { "balance" : { "order" : "desc" } }
}
```

검색 결과로 원하는 항목만 추출

```
{
  "query" : { "match_all" : {} },
  "_source" : [ "balance", "account_number" ]
}
```

```
{
  "_index": "bank",
  "_type": "account",
  "_id": "25",
  "_score": 1,
  "_source": {
    "account_number": 25,
    "balance": 40540
  }
},
{
  "_index": "bank",
  "_type": "account",
  "_id": "44",
  "_score": 1,
  "_source": {
    "account_number": 44,
    "balance": 34487
  }
}
```

Match query

(post) localhost:9200/bank/_search?pretty

```
{
  "query" : {"match" : {"account_number" : 20 }},
  "_source" : ["balance", "firstname"]
}
```

```
'hits': [
  {
    "_index": "bank",
    "_type": "account",
    "_id": "20",
    "_score": 1,
    "_source": {
      "firstname": "Elinor",
      "balance": 16418
    }
  }
]
```

Address에 mill 또는 lane을 포함한 문서

```
{
  "query" : {"match" : {"address" : "mill lane"}}
}
```

```
'hits': [
  {
    "_index": "bank",
    "_type": "account",
    "_id": "136",
    "_score": 7.3900023,
    "_source": {
      "account_number": 136,
      "balance": 45801,
      "firstname": "Winnie",
      "lastname": "Holland",
      "age": 38,
      "gender": "M",
      "address": "198 Mill Lane",
      "employer": "Neteria",
      "email": "winnieholland@neteria.com",
      "city": "Urie",
      "state": "IL"
    }
  }
],
```

bool 이용

address에 mill lane 모두 포함한 문서 (bool must)

```
{
  "query" : {
    "bool" : {
      "must" : [
        {"match" : {"address" : "mill"}},
        {"match" : {"address" : "lane"}}
      ]
    }
  }
}
```

```
"_source": {
  "account_number": 136,
  "balance": 45801,
  "firstname": "Winnie",
  "lastname": "Holland",
  "age": 38,
  "gender": "M",
  "address": "198 Mill Lane",
  "employer": "Neteria",
  "email": "winnieholland@neteria.com",
  "city": "Urie",
  "state": "IL"
}
```

Address에 mill 또는 lane 포함한 문서 (bool should)

```
{
  "query" : {
    "bool" : {
      "should" : [
        {"match" : {"address" : "mill"}},
        {"match" : {"address" : "lane"}}
      ]
    }
  }
}
```

```
"_source": {
  "account_number": 136,
  "balance": 45801,
  "firstname": "Winnie",
  "lastname": "Holland",
  "age": 38,
  "gender": "M",
  "address": "198 Mill Lane",
  "employer": "Neteria",
  "email": "winnieholland@neteria.com",
  "city": "Urie",
  "state": "IL"
}
```


Address에 mill lane 둘다 포함 x (must_not)

```
{
  "query" : {
    "bool" : {
      "must_not" : [
        {"match" : {"address" : "mill"}},
        {"match" : {"address" : "lane"}}
      ]
    }
  }
}
```

```
{
  "_index": "bank",
  "_type": "account",
  "_id": "25",
  "_score": 1,
  "_source": {
    "account_number": 25,
    "balance": 40540,
    "firstname": "Virginia",
    "lastname": "Ayala",
    "age": 39,
    "gender": "F",
    "address": "171 Putnam Avenue",
    "employer": "Filodyne",
    "email": "virginiaayala@filodyne.com",
    "city": "Nicholson",
    "state": "PA"
  }
}
```

모두 동시에 사용 가능

```
{
  "query": {
    "bool": {
      "must": [
        { "match": { "age": "40" } }
      ],
      "must_not": [
        { "match": { "state": "ID" } }
      ]
    }
  }
}
```

필터링

_score : 검색 조건에 얼마나 부합한지에 대한 수치

range 쿼리 : 검색된 문서를 필터링

```
{
  "query" : {
    "bool" : {
      "must" : { "match_all" : {} },
      "filter" : {
        "range" : {
          "balance" : {
            "gte" : 20000,
            "lte" : 30000
          }
        }
      }
    }
  }
}
```

```
"_index": "bank",
"_type": "account",
"_id": "253",
"_score": 1,
"_source": {
  "account_number": 253,
  "balance": 20240,
  "firstname": "Melissa",
  "lastname": "Gould",
  "age": 31,
  "gender": "M",
  "address": "440 Fuller Place",
  "employer": "Buzzopia",
  "email": "melissagould@buzzopia.com",
  "city": "Lumberton",
  "state": "MD"
}
```

집계

Size = 0이니까 집계 결과만 출력하고

State로 묶어서

```
{
  "size" : 0,
  "aggs" : {
    "group_by_state" : {
      "terms" : {
        "field" : "state.keyword"
      }
    }
  }
}
```

```
"aggregations": {
  "group_by_state": {
    "doc_count_error_upper_bound": 20,
    "sum_other_doc_count": 770,
    "buckets": [
      {
        "key": "ID",
        "doc_count": 27
      },
      {
        "key": "TX",
        "doc_count": 27
      },
      {
        "key": "AL",
        "doc_count": 25
      }
    ]
  }
}
```

상위 10개 주의 평균 잔액계산

```
{
  "size" : 0,
  "aggs" : {
    "group_by_state" : {
      "terms" : {
        "field" : "state.keyword"
      },
      "aggs" : {
        "average_balance" : {
          "avg" : {
            "field" : "balance"
          }
        }
      }
    }
  }
}
```

```
'buckets': [
  {
    "key": "ID",
    "doc_count": 27,
    "average_balance": {
      "value": 24368.777777777777
    }
  },
  {
    "key": "TX",
    "doc_count": 27,
    "average_balance": {
      "value": 27462.925925925927
    }
  },
  {
    "key": "AL",
    "doc_count": 25,
    "average_balance": {
      "value": 25739.56
    }
  }
]
```

Group_by_state, average_balance는 이름