ElasticSearch에 대해 알아보자.txt

당근마켓
SoftwareEngineer
dydwls121200@gmail.com
(2019.04 ~) 조 용진

당근마켓 (2019.04 ~)

플랫폼개발팀

모두의캠퍼스 (2017.05 ~ 2019.03)

- 나는 개발잘하니까 들어갔는데, 생각보다 쉽지 않음
- 현실은 만만하지 않았음. 사장은 잘했는데 내가 부족했음
- 사장과 함께 당근마켓 랜딩

모빌C&C (2016.07 ~ 2017.01)

- 대학생활동안 SI Stack에 맞는 공부함
- 상상속 SI와 현실은 달랐다.
- 정직원 할까 말까 하다 괴리감에 후퇴

가천대 (2011.03 ~ 2017.08)

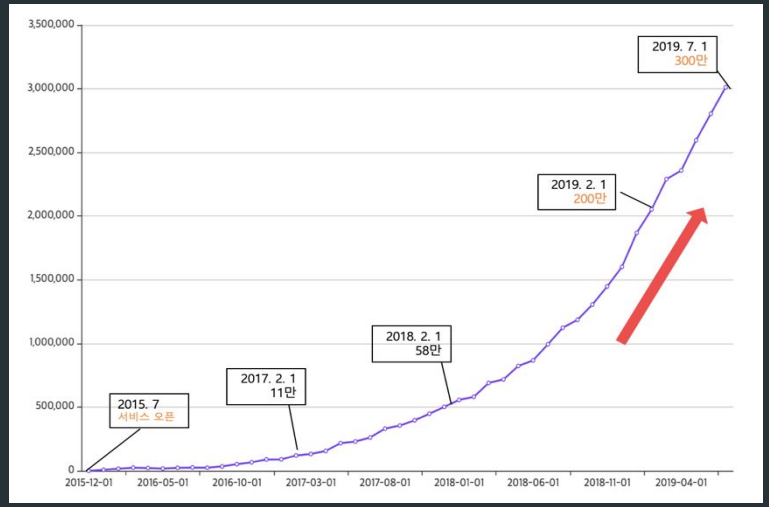
- 기타에 이 한몸 바쳤~~음(사실 엄청 못침~~)
- 기타칠때만, 술, 담배, 음악이~~면 되는 히피같~~은 생활 함
- 군대 갔다 와서 정신차림(군버프는 실존했다)

각설

ElasticSearch는 뭐하는 녀석일까?

ElasticSearch는 정말 잘 만든 엔진이다.

당근마켓의 검색에는
ElasticSearch를 사용한다.



ElasticSearch를 4년 넘게 써오고 있지만

엔진 자체에 뭔가 이렇다할 장애가 없었다.

(물론, ~~뭔가 특별히 커타란 뭔가를 허지 않기도 했다~~.)

그러므로, ElasticSearch는 정말 잘 만든 엔진이다.

(~~클러스터링과 샤딩 안정성 ㅆㅅㅌㅊ~~)

ElasticSearch는 다들 알겠지만
로그분석 및 검색의 역할로 사용한다.

이번 발표에서는 ElasticSearch에 대해서 알아보려 한다.

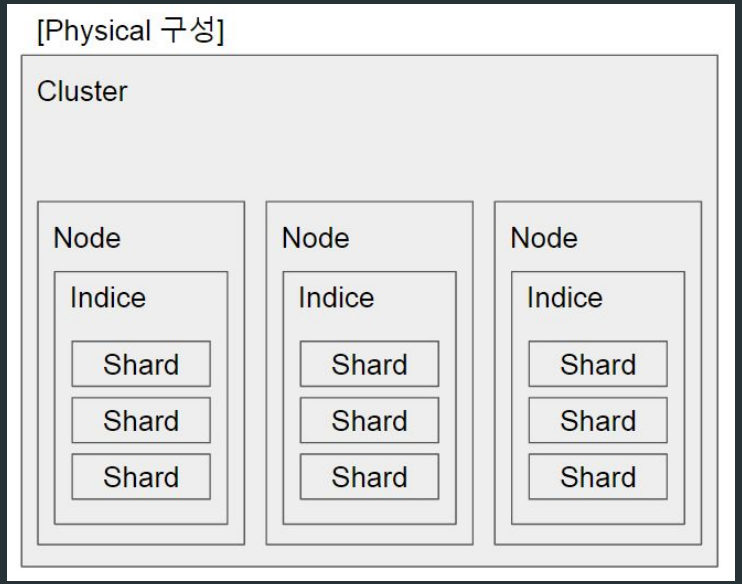
ElasticSearch는 검색엔진으로 좋나요?

검색에 대한 도메인은 없고, 러닝커브가질 시간도 없는데
like search(ngram)으로 충분하다면 Mysql fulltext, AWS cloud search를
쓰세요.

러닝커브 타임이 있어도 괜찮고, 장기적으로 고도화를 할 계획이라면
ElasticSearch 추천해요.

~~검색경력이 수 년 가까이되고, 산전수전 겪으신 분이라면..~~
저 좀 알려주세요. 아니면 당근마켓의 동료가 되는건 어때요...?

간단히 짚고 넘어가는 ElasticSearch

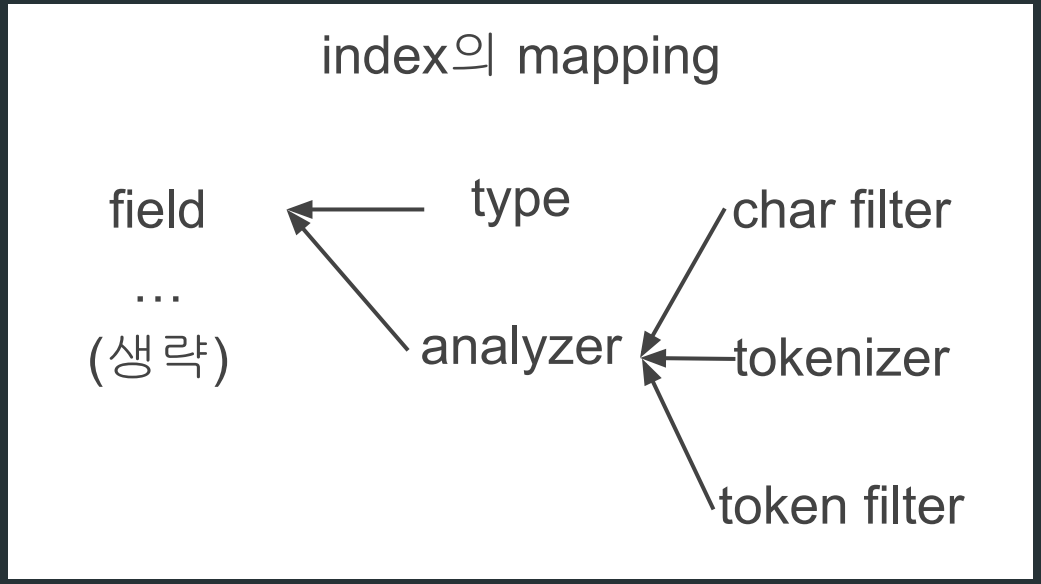


ElasticSearch는 여러개의 index를 가질 수 있고 RDBMS에서는 Table,
NoSQL에서는 collection과 같은 개념이다.

ElasticSearch는 클러스터를 구성해서 운영하고,
클러스터 회복 및 확장에 많은 기능을 제공한다.

Index를 만들때에는 mapping이라는것을 정의 해 주어야 하는데
schema정의와 같다

mapping에서는 field라는 속성이 있는데,
이 속성 마다의 설정이 검색성능에 중요한 factor가 된다.



1개의 mapping field를 위한 설정은 type과 analyzer로 설정하며 analyzer를 만들기위해서는 char filter, tokenizer, token filter를 조합해서 만들어야 한다.

[Source]

↓

n개의 Char. Filter

↓

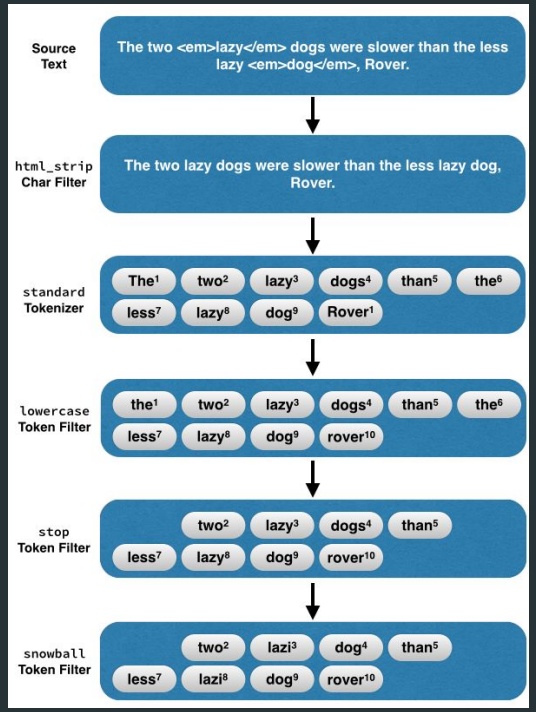
1개의 Tokenizer

↓

n개의 Token Filter

↓

[Output]



Analyzer는 term을 추출하는 녀석이며,
저장과 검색시 term을 추출할 때 다음과 같이
동작한다.

응? 텀? term?

term = 색인어(검색에 사용되는 단어)

한개의 문서에 '블랙핑크'가 '블랙핑크'는 '블랙핑크'와 '블랙핑크'를
*가, 는, 와, 를 상관없이 검색당할 **factor** 가 바로 **term***

“Analyzer는 term을 추출하는 녀석이며,
저장과 검색시 term을 추출할 때 다음과 같이 동작한다.”

왜 저장과 검색시에 term을 추출해야해?

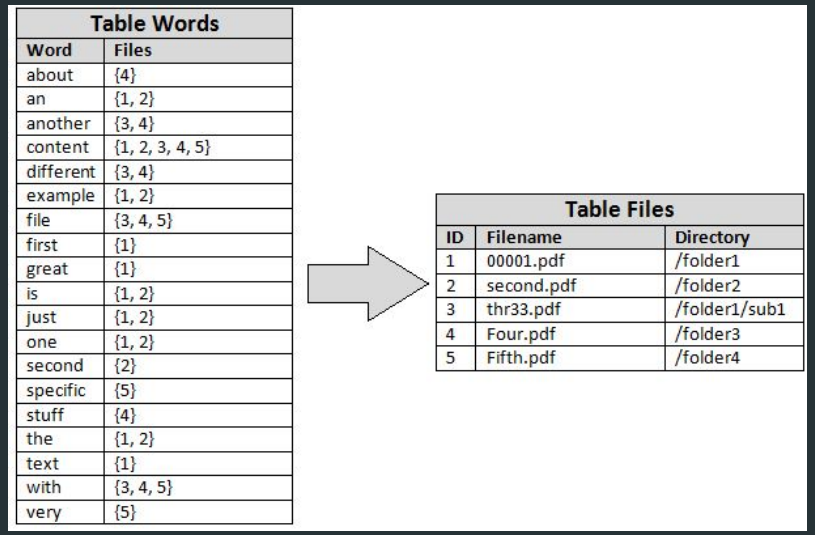


Table Words	
Word	Files
about	{4}
an	{1, 2}
another	{3, 4}
content	{1, 2, 3, 4, 5}
different	{3, 4}
example	{1, 2}
file	{3, 4, 5}
first	{1}
great	{1}
is	{1, 2}
just	{1, 2}
one	{1, 2}
second	{2}
specific	{5}
stuff	{4}
the	{1, 2}
text	{1}
with	{3, 4, 5}
very	{5}

Table Files		
ID	Filename	Directory
1	00001.pdf	/folder1
2	second.pdf	/folder2
3	thr33.pdf	/folder1/sub1
4	Four.pdf	/folder3
5	Fifth.pdf	/folder4

Inverted Index란 것을 만들어야 합니다.
데이터 저장시에 term(word)을 추출해서 문서와 연결을 짓는 vector space입니다.

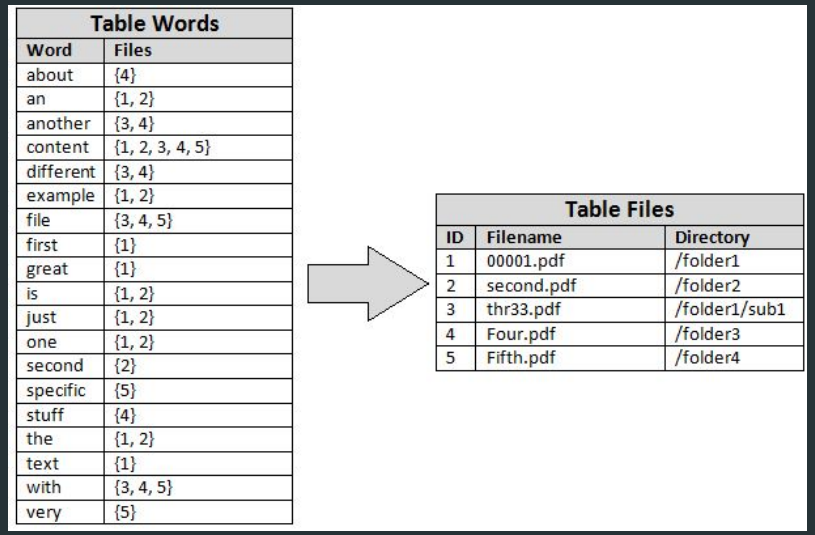


Table Words	
Word	Files
about	{4}
an	{1, 2}
another	{3, 4}
content	{1, 2, 3, 4, 5}
different	{3, 4}
example	{1, 2}
file	{3, 4, 5}
first	{1}
great	{1}
is	{1, 2}
just	{1, 2}
one	{1, 2}
second	{2}
specific	{5}
stuff	{4}
the	{1, 2}
text	{1}
with	{3, 4, 5}
very	{5}

Table Files		
ID	Filename	Directory
1	00001.pdf	/folder1
2	second.pdf	/folder2
3	thr33.pdf	/folder1/sub1
4	Four.pdf	/folder3
5	Fifth.pdf	/folder4

검색어에 term(word) 중에 Inverted Index에
해당하는 문서가 있으면 출력할 수 있습니다.

그러므로, Analyzer는 Inverted Index에 저장될 term의 형태를 정의한다.

Analyzer는 term을 추출한다. 그렇다면 한글은 어떻게...?

“아버지가방에들어가신다”, “사람 주차장”, “지하 자동차 주차장”
‘블랙핑크’가 ‘블랙핑크’는 ‘블랙핑크’와 ‘블랙핑크’를

*영어는 **whitespace**기준으로 **term**을 뽑고... 전치사 대명사 등 빼면 유의미한 단어가 남아도..*
한글은? 어떻게? 뽑지? 가,는,와,를

형태소 분석기

nori, mecab, kolnpy, korea-open-text,
eunjoen, seunjeon, arirang

어디서 들어본거는 같은데, 이건 뭘까?

ElasticSearch 입장에서 보면 형태소분석기는
품사에 따라 단어들을 쪼개주는 역할을 한다.
Tokenizer!

“아버지가방에들어가신다” => “아버지” “가” “방” “에” “들어가” “신다”

품사 뿐만 아니라, 복합명사도
쪼개버린다.

‘이사업체’ => ‘이사’, ‘업체’

만약 안쪼개진다면?

쇼핑몰에서 “갤럭시핸드폰” 이라 검색하면
“갤럭시”, “핸드폰” 으로 분리가 안되어서 “갤럭시핸드폰” 이라는 term을 찾게됨

엄연히 “갤럭시”와 “갤럭시핸드폰”은 inverted index vector space에서는
부분집합도 뭣도 아닌 다른 term임.
때문에, “갤럭시” term이 포함된 상품이 안나오게 됨

이런 경우에는 '사용자사전'을 이용해서 문제를 해결하도록 한다.

"{복합명사}{복합}{명사}"
=> "갤럭시핸드폰 갤럭시 핸드폰"

[아이폰, 케이스]

[아이, 폰, 케이스]

[아이폰케이스]

사용자사전은 명사의 **term**을 어떻게 분리 시킬지에 대한 정의를 하는 곳

가끔 이런 문제도 있다

아이폰이나 iphone이나 IPHONE이나 애플폰이나 에플폰이나

모두 사이좋은 아이폰이다.

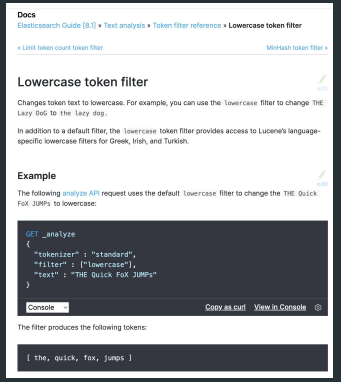
“IPHONE+로 검색해도

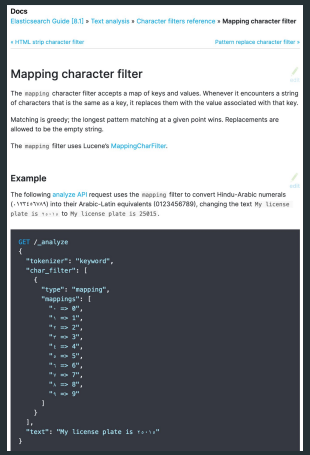
아이폰/ iphone/ IPHONE/ 애플폰/ 에플폰/ 아이폰 플러스/ 아이폰+

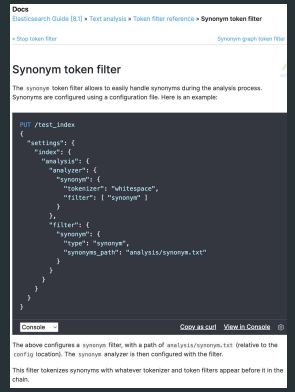
나오게 해줘”

이 대부분의 요건들을 충족시킬 수 있는것이 바로

elasticsearch의 analyzer기능에 붙어있는 플러그인들







전부 lowercase로 싹 밀어버리고, (lowercase filter)

+ 는 plus라는 단어로 replace시키도록 하고, (mapping char. filter)

아이폰 = 애플폰, iphone,에플폰 이라는 처리 해버리고 (synonym filter)

오타제거는 머신러닝모델이나 통계모델써서 해결하거나..

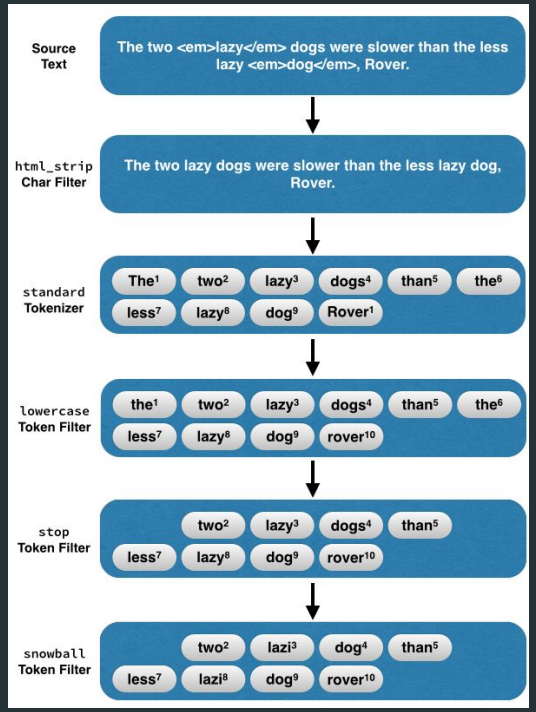
[Source]

n개의 Char. Filter

1개의 Tokenizer

n개의 Token Filter

[Output]



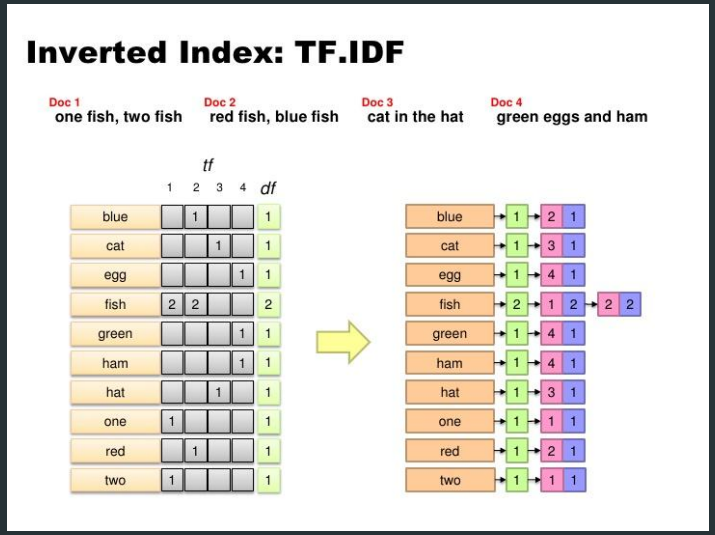
analyzer를 어떻게 구성하느냐에 따라
원하는 문서를 가져오는 연관도가 달라지게 된다.

case1, “엿” 만 검색해도 “역삼역”이 나왔으면 좋겠어 (모바일환경)
case2, “durtka” 을 검색해도 “역삼역”이 나왔으면 좋겠어 (한영키보드 배치)
case3, “보라엄마”검색해도 “강보라엄마” 가 나왔으면 좋겠어 (일부분만 기억나는 경우)
case4, “메리크리스마스” 검색해도 “매리크리스마스”가 나왔으면 좋겠어 (오타)
case5, 초성으로 검색할 수 있으면 좋겠어 (노래방)
…

그러므로 색인어 term을 만드는
Char.Filter / Tokenizer / Token Filter의 도구들을 여럿 알고 있는게 도움이 된다.

matching만큼은 어떻게 해야할지 조합해서 만들 수 있다.

term을 갖고 있는 문서들을 출력했지만, sorting은 제 멋대로다.
때문에 문서들 간에 ranking결정에 사용하는것이 바로 TF-IDF



Term Frequency - Inverse Document Frequency

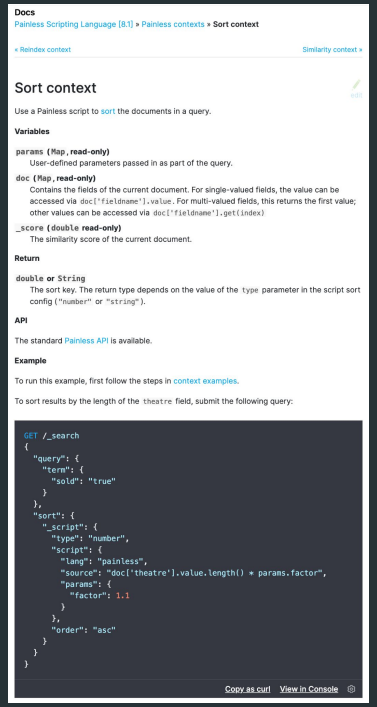
TF - 문서 당 특정 단어의 빈도가 높으면 해당 단어는 문서를 대표하는 단어라 생각한다

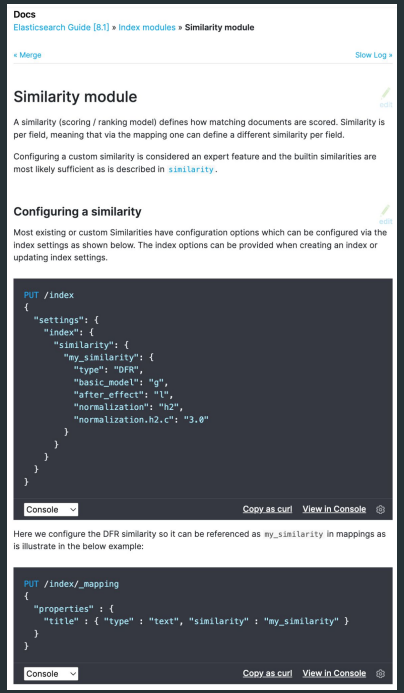
IDF - 문서군 사이에서 자주 등장(DF)하는 단어면 중요하지 않은단어기 때문에 inverse 값을 이용한다.

```
{
    "_index": "some_index",
    "_type": "_doc",
    "_id": "12345",
    "matched": true,
    "explanation": {
        "value": 0.2876821,
        "description": "weight(string field:object in 0) [PerFieldSimilarity], result of:",
        "details": [
            {
                "value": 0.2876821,
                "description": "score(freq=1.0), product of:",
                "details": [
                    {
                        "value": 2.2,
                        "description": "boost",
                        "details": []
                    },
                    {
                        "value": 0.2876821,
                        "description": "idf, computed as log(1 + (N - n + 0.5) / (n + 0.5)) from:"
                        "details": [
                            {
                                "value": 1,
                                "description": "n, number of documents containing term",
                                "details": []
                            },
                            {
                                "value": 1,
                                "description": "N, total number of documents with field",
                                "details": []
                            }
```


ElasticSearch의 ranking score에는

기본적으로 TF-IDF를 이용한다.(\w BM25)





ElasticSearch에서는 sorting 에 대해 제어할 수 있게 api들을 제공하며, 다음 문서들을 참고할 수 있다.

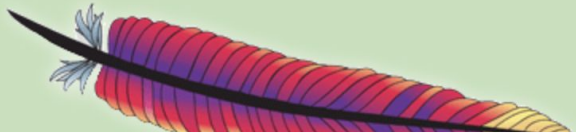
문서를 찾는 것과 정렬하는 것에 대해서 알아보았다.
그렇다면 이것은 물리적으로 누가 하는가?

Search with Apache So @ select provider

Lucene™

Core (Java) | Solr | PyLucene

Apache 2.0 licensed



Lucene Engine

Welcome to Apache Lucene

The Apache Lucene™ project develops open-source search software, including:

- Lucene Core, our flagship sub-project, provides Java-based indexing and search technology, as well as spellchecking, hit highlighting and advanced analysis/tokenization capabilities.
- Solr™ is a high performance search server built using Lucene Core, with XML/HTTP and JSON/Python/Ruby APIs, hit highlighting, faceted search, caching, replication, and a web admin interface.
- PyLucene is a Python wrapper around the Core project.

DOWNLOAD

Apache Lucene 8.4.1

DOWNLOAD

Apache Solr 8.4.1

Projects

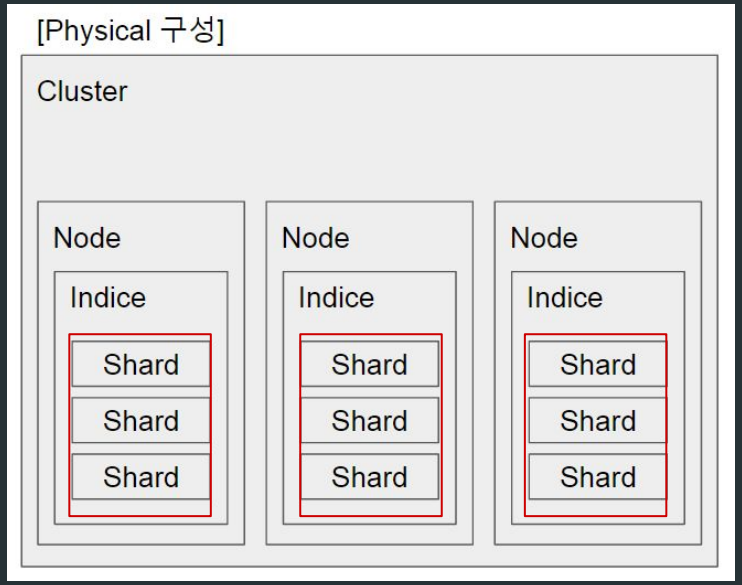
Lucene Core (Java)

Lucene

Elasticsearch에서 실질적으로 검색을 수행해주는 녀석

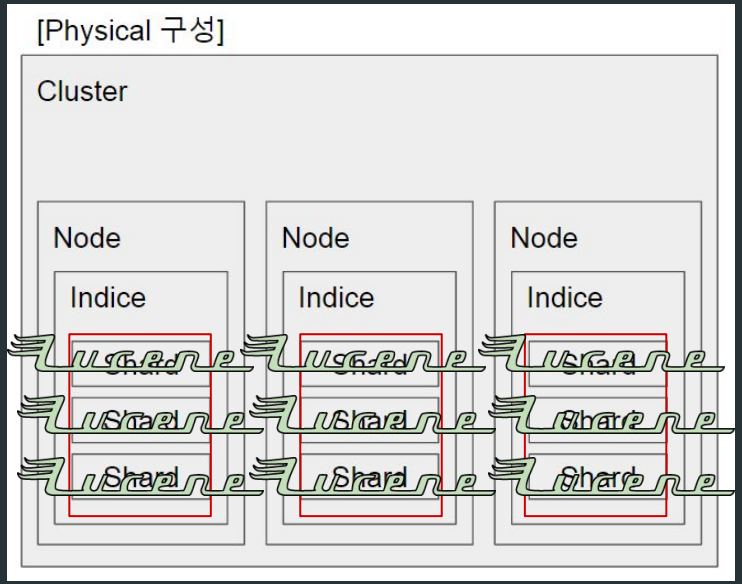
?

Elasticsearch에 Lucene이 어디있어?


[Physical 구성]
Cluster
Node
Indice
Shard
Shard
Shard
Node
Indice
Shard
Shard
Shard
Node
Indice
Shard
Shard
Shard

샤드

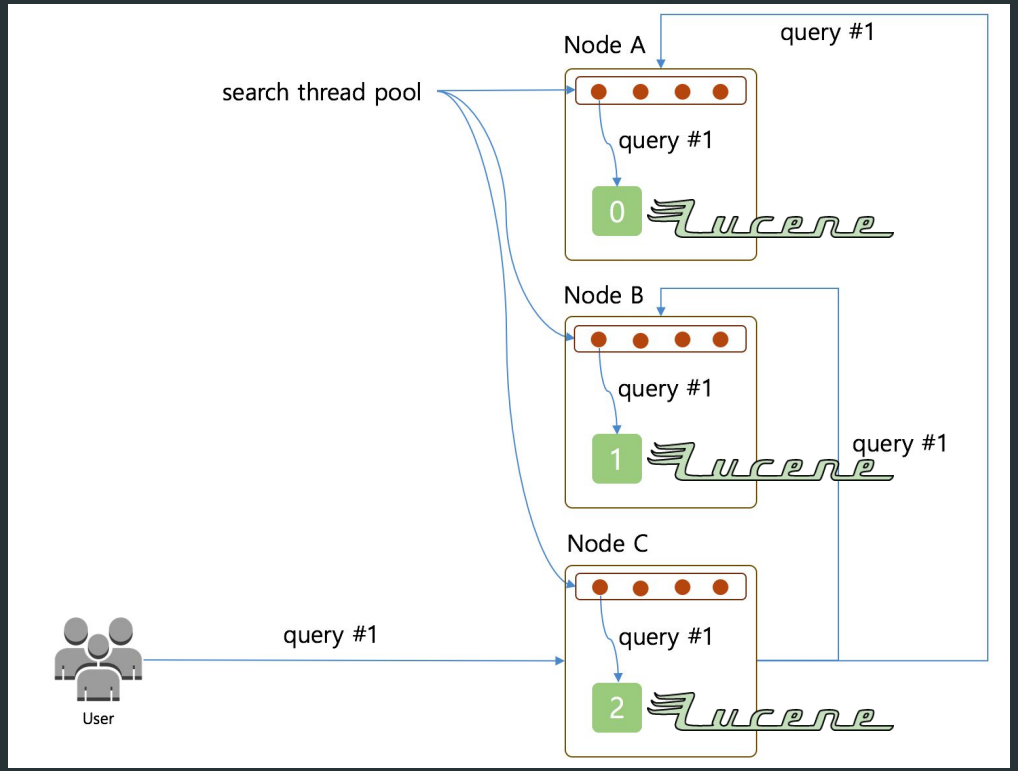
(Shard)



샤드(shard) => Lucene

Shard는

데이터베이스 샤드는 데이터베이스나 웹 검색 엔진의 데이터의 수평 분할이다.
개개의 파티션은 샤드 또는 데이터베이스 샤드로 부른다.
각 샤드는 개개의 데이터베이스 서버 인스턴스에서 부하 분산을 위해 보유하고 있다.



출처 - 클러스터 설계하기 - #1 검색 성능과 샤드 개수 by 강진우

검색은 Lucene이 해주는데
ElasticSearch는 뭐 해?

주로
샤드관리, 클러스터관리, 각종형분기,
각종플러그인(보안, 유틸, ML detection, APM, analyzer)

하지만 사실, 검색은 lucene만 하지 않음. elasticsearch도
lucene검색할때에 유용한 도구들을 많이 제공하는 편

그래 검색과 ElasticSearch에 대한건
어느~~정도는 알게되었다.

나머지는 실무로 뚜드려 맞다보면 알게된다.

우리인생 화이팅 :D

끝!

3분.엘라스틱서치

https://velog.io/@drakejin/3%EB%B6%84.%EC%97%98%EB%9D%BC%EC%8A%A4%ED%8B%B1%EC%84%9C%EC%B9%98.txt