엘라스틱서치

오늘의 학습내용

- 엘라스틱서치(Elasticsearch) 기본 개념
- 엘라스틱서치(Elasticsearch) 구성도
- REST API

엘라스틱서치(Elasticsearch) 기본 개념

Elasticsearch 기본 개념

- Near Realtime
- Cluster : 전체 데이터를 함께 보유하고 모든 노드에서 연합 인덱싱 및 검색 기능을 제공하는 하나 이상의 노드 모음
- Node : 클러스터의 일부이며 데이터를 저장하고 클러스터의 인덱싱 및 검색 기능에 참여하는 단일 서버
- Index : 다소 유사한 특성을 갖는 문서들의 집합
- Type : Index 내에서 하나 이상의 Type을 정의
- Document : Index를 생성 할 수 있는 기본 정보 단위, JSON표현
- Shards : shards를 이용하여 Index를 여러 조각으로 분할 가능, 기본 4개 제공
- Replication : 장애가 발생할 경우 고가용성을 제공, 기본 1개 제공

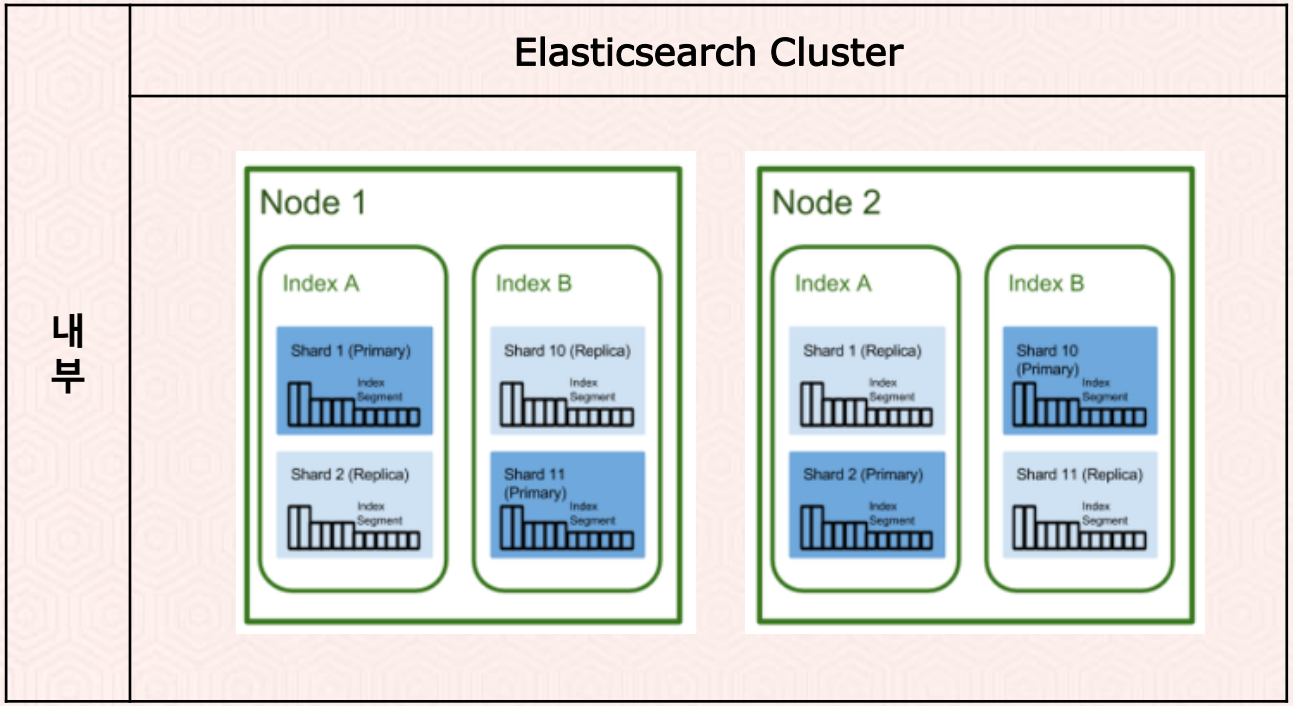
엘라스틱서치(Elasticsearch) 기본 개념

- 관계형 데이터베이스와 Elasticsearch 용어 비교

관계형 데이터베이스	Elasticsearch
Database	Index
Table	Type
Row	Document
Column	Field
Schema	Mapping
Index	Everything is indexed
SQL	Query DSL

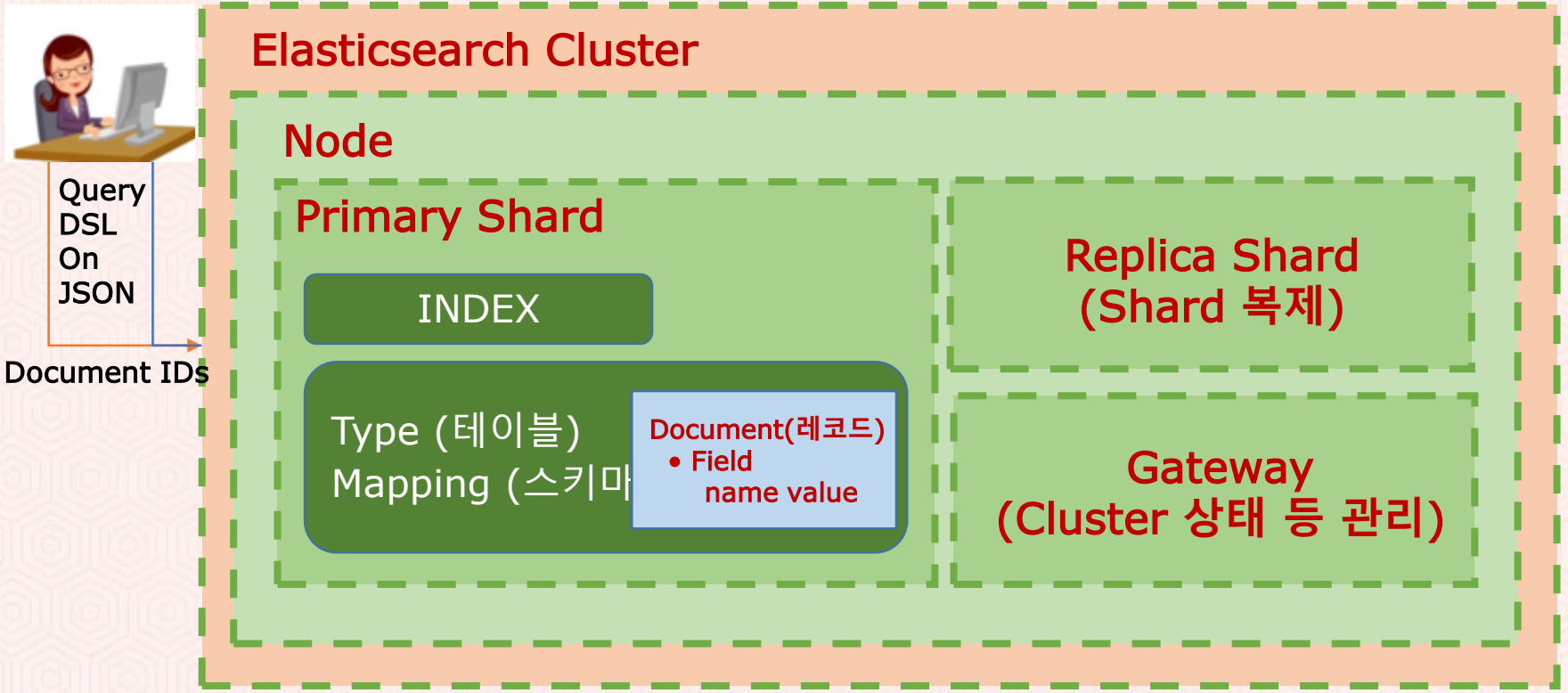
엘라스틱서치(Elasticsearch) 구성도

- 아키텍쳐



엘라스틱서치(Elasticsearch) 구성도

개념적 구성도



엘라스틱서치(Elasticsearch) 구성도

REST API 기본 구조

- Elasticsearch는 REST (Representational State Transfer) API를 제공하여 다양한 환경에서 사용 가능
- REST에서 Methods의 주요 용도
  - POST : 등록 (Create)
  - PUT : 수정 (Replace), 데이터가 없을 경우에는 등록 (Create)
  - DELETE : 삭제 (Delete)
  - GET : 조회 (List, Retrieve)

REST API

URI 기본 형태

- http://localhost:9200/index/type/id?parameters
- http://localhost:9200/[index/[type/]action?parameters]

-index : DBMS에서 데이터베이스에 해당

-type : DBMS에서 테이블에 해당

-id : DBMS에서 레코드에 해당하는 Document의 ID

-index, type, id를 여러 개 지정할 경우 ","를 사용하여 구분

"*"를 사용하여 모두 지정 가능

- action : 특정 작업을 지시

- 공통 parameters (pretty, v, help, h=컬럼1,컬럼2, bytes=b)

REST API

Action 1

Action	상세
_cluster	클러스터 관련 작업
_nodes	노드 관련 작업
_aliases	Index alias 관련 작업
_analyze	analyzer 관련 작업
_cache	Cache 관련 작업
_flush	Transaction log 또는 Memory free 작업
_optimize	Segment 파일 병합 작업
_stats	시스템 또는 색인의 통계 정보
_search	검색 작업
_msearch	Multi 검색 작업

REST API

Action 2

Action	상세
_mget	Multi Document patch 작업
_validate	Query에 대한 유효성 검사 작업
_suggest	검색어 자동 완성
_bulk	Bulk 색인 작업
_count	문서 count 작업
_settings	elasticsearch.yml에 설정한 global settings 정보 조회 http://localhost:9200/index/_settings?pretty=true number_of_shards : Shard 개수 number_of_replicas : Replica 개수 index.refresh_interval : Index 변경 후 검색 결과에 반영되는 시간 설정 analysis :analyzer와 tokenizer 설정 store : 저장 옵션

REST API

Mapping 1

Attribute	Default	상세
store	no	원본 저장 여부
index	analyzed	색인 방식 지정 no, not_analyzed, analyzed
term_vector	no	색인어에 대한 메타 정보 저장 방식 yes, no, with_offsets, with_positions, with_positions_offsets
boost	1.0	Boost 값
null_value		필드의 값이 null일 경우의 default 값
omit_norms	true	Lucene의 norms 사용 여부
index_options	positions	색인시 저장할 메타 정보 설정 positions, docs

REST API

Mapping 2

Attribute	Default	상세
analyzer		색인 및 검색 시 사용할 Global analyzer
index_analyzer		색인 시 사용할 analyzer
search_analyzer		검색 시 사용할 analyzer
include_in_all	true	_all 필드에 검색 가능한 모든 필드를 포함할지 여부
ignore_above		문자열 필드에서 정해진 크기를 넘는 문자는 무시하도록 설정
position_offset_gap		Phrase 검색에서 전후 텍스트간의 간격 설정
precision_step		최대 number_term 설정
ignore_malformed	false	잘못된 number, date 무시
format	dateOptionalTime	Date format

REST API

REST API 등록/수정/삭제/조회

등록 (POST/PUT)	•**customer 인덱스 생성** curl -XPUT 'node201.hadoop.com:9200/customer?pretty' curl -GET 'node201.hadoop.com:9200/_cat/indices?v' •**External 타입으로 문서 추가** •문서 번호는 자동으로 생성 curl -XPOST 'node201.hadoop.com:9200/customer/external?pretty' -d ' { "name": "Mountain Lover" }' curl -XGET 'node201.hadoop.com:9200/customer/external/1lz2jL6CQui07FnZGd_R9w?pretty' •**External 타입으로 1번 문서 추가** curl -XPUT 'node201.hadoop.com:9200/customer/external/1?pretty' -d ' { "name": "Mountain Lover" }' curl -XGET 'node201.hadoop.com:9200/customer/external/1?pretty'

REST API

REST API 등록/수정/삭제/조회

수정 (PUT/POST)	**•external 타입으로 1번 문서 수정** curl -XPOST 'node201.hadoop.com:9200/customer/external/1/_update?pretty' -d '{ "doc": { "name": "Mountain Lover!", "age": 20 } }' curl -XGET 'node201.hadoop.com:9200/customer/external/1?pretty' **•external 타입으로 1번 문서 수정** curl -XPUT 'node201.hadoop.com:9200/customer/external/1?pretty' -d ' { "name": "Mountain Lover!" }' curl -XGET 'node201.hadoop.com:9200/customer/external/1?pretty'
삭제 (DELETE)	**•문서 삭제** curl -XDELETE 'node201.hadoop.com:9200/customer/external/1?pretty' curl -XGET 'node201.hadoop.com:9200/customer/external/1?pretty' curl -XDELETE 'node201.hadoop.com:9200/customer/external/_query?pretty' -d ' { "query": { "match": { "name": "Mountain Lover!" } } }' curl -XGET 'node201.hadoop.com:9200/customer/external/1?pretty' **•customer 인덱스 삭제** curl -XDELETE 'node201.hadoop.com:9200/customer?pretty' curl -GET 'node201.hadoop.com:9200/_cat/indices?v'

REST API

REST API 등록/수정/삭제/조회

조회 (GET)	•**조회** curl -GET 'node201.hadoop.com:9200/_cat/indices?v' curl -XGET 'node201.hadoop.com:9200/customer/external/1?pretty' •**조회되는 데이터 구조** _index _type _id _version : 1, 2, 3, ... _source : { name1: value1, name2: value2 }
검색 (GET/POST)	•**REST request URI** curl -XGET 'node201.hadoop.com:9200/customer/_search?q=*&pretty' •**REST request body** curl -XPOST 'node201.hadoop.com:9200/customer/_search?pretty' -d ' { "query": { "match_all": {} } }'

REST API

Document API

- index api : -XPUT : 등록, -XPOST : 등록 (id 자동 생성)
- get api : -XGET
- delete api : -XDELETE
- update api : -XPUT
- multi get api : -XGET, /_mget

REST API

Search API

- q : Query String Query
- analyzer
- default_operator : OR (default), AND
- _source = false, _source_include, _source_exclude
- df : 디폴트 필드 지정
- fields : 필드 지정
- sort : field:asc, field:desc

REST API

Basic API – Index 관리

- curl -X GET http://node201.hadoop.com:9200/_status #--- 상태 확인
- curl -X POST http://node201.hadoop.com:9200/index001 #--- index 생성
- curl -X DELETE http://node201.hadoop.com:9200/index001 #--- index 삭제
- curl -X GET http://node201.hadoop.com:9200/index001/_mapping #--- Mapping 조회
- curl -X GET http://node201.hadoop.com:9200/index001/_status #--- 상태 확인
- curl -X GET http://node201.hadoop.com:9200/index001/_search #--- 검색
- curl -X GET http://node201.hadoop.com:9200/_all/_search #--- 검색

REST API

Basic API – 테이블 관리

- curl -X POST http://node201.hadoop.com:9200/index001/type001 -d '{ title: "Greeting", body: "Hello World!" }'
- curl -X DELETE http://node201.hadoop.com:9200/index001/type001 #--- type 삭제
- curl -X GET http://node201.hadoop.com:9200/index001/type001/_mapping #--- Mapping 조회
- curl -X GET http://node201.hadoop.com:9200/index001/type001/_status #--- 상태 확인
- curl -X GET http://node201.hadoop.com:9200/index001/type001/_search #--- 검색
- http://node201.hadoop.com:9200/index001/type001/_search?q=title:Gre*ting

REST API

Basic API – Mapping 관리

❖ #--- Mapping 생성

```
curl -X PUT http://node201.hadoop.com:9200/index001/type001/_mapping -d '{
  type001: {
   properties: {
     title: {
       type: "string",
       index: "not_analyzed"
     } …..
curl -X GET http://node201.hadoop.com:9200/index001/type001/_mapping   #---
Mapping 조회
```

REST API

Basic API – Document 관리 (레코드)

- curl -X POST http://node201.hadoop.com:9200/index001/type001/data001 -d '{ title: "Greeting", body: "Hello World!" }'
- curl -X POST http://node201.hadoop.com:9200/index001/type001/data001/_update -d '{ title: "Greeting", body: "Hello World!" }'
- curl -X DELETE http://node201.hadoop.com:9200/index001/type001/data001 #--- data001 데이터 삭제
- curl -X GET http://node201.hadoop.com:9200/index001/type001/data001 #--- data001 데이터 조회
- curl -X GET http://node201.hadoop.com:9200/index001/type001/_search -d '{query: {text: {_all: "Hello"}}}`#--- document 검색

REST API

Bulk API

- Bulk로 문서 등록, 수정, 삭제
- curl -XPOST 'node201.hadoop.com:9200/customer/external/_bulk?pretty' -d '

```
{"index":{"_id":"1"}}
{"name": "John Doe" }
{"index":{"_id":"2"}}
{"name": "Jane Doe" }
```