챕터4 데이터 관리를 위한 엘라스틱서치 기초

Created 2021년 11월 28일 오전 10:44

Tags 비어 있음

엘라스틱서치 소개와 설치

```
sudo apt update && sudo apt install -y docker.io docker network create elastic
docker run -d --name es01-test --net elastic -p 9200:9200 -p 9300:9300 -e "dis
covery.type=single-node" elasticsearch:7.14.1 docker run -d --name kib01-test
--net elastic -p 5601:5601 -e "ELASTICSEARCH_HOSTS=http://es01-test:9200" kiba
na:7.14.1
```

도큐먼트 입력과 조회

인덱스 조회하기

```
GET /_cat/indices?v
```

인덱스 생성하기

```
PUT /customer?pretty GET /_cat/indices?v <curl 명령어> curl -X PUT "localhost:9
200/customer?pretty" curl -X GET "localhost:9200/_cat/indices?v"
```

인덱스에 도큐먼트 입력

```
POST customer/_doc/1 { "name": "John Doe" } GET customer/_doc/1
```

인덱스 삭제

```
DELETE /customer?pretty GET /_cat/indices?v
```

여러 개 필드 입력하기 (인덱스 자동 생성)

```
POST books/_doc/1 { "title": "Elasticsearch Guide", "author": "Choi", "date" :
"2021-10-30", "pages" : 500 }
```

도큐먼트만 삭제하기

```
DELETE /customer/_doc/2?pretty
```

데이터 삭제 시 특징

- 메타데이터가 그대로 유지
- 도큐먼트 삭제 후 다시 데이터를 입력하면 _version 값이 이어서 진행
- 버전까지 초기화하려면 인덱스를 삭제해야 함

도큐먼트 업데이트

데이터 교체하기

```
POST /customer/_doc/2?pretty { "name": "Jane Doe" }
```

스크립트를 활용한 데이터 업데이트

```
POST customer/_doc/1 { "name": "John Doe" } POST customer/_update/1 { "doc" :
{ "category" : "IT", "pages" : 50 } } POST customer/_update/1 { "doc" : { "aut
hor" : "CHOI" } }
```

스크립트를 활용한 데이터 업데이트

```
POST customer/_update/1 { "script" : "ctx._source.pages+=50" } GET customer/_d
oc/1
```

스크립트를 활용한 데이터 삭제

```
POST customer/_update/1/ { "script" :{ "source" : "if(ctx._source.pages <= par
ams.page_cnt){ctx.op='delete'} else{ctx.op='none'}", "params": { "page_cnt" :
150 } } }
```

데이터 입력과 조회 연습문제

TourCompany의 고객관리를 위해 다음 데이터를 입력하십시오. Index 이름은 tourcompany로 사용

Doc Id	name	phone	holiday_dest	departure_date
1	Alfred	010-1234-5678	Disneyland	2017/01/20
2	Huey	010-2222-4444	Disneyland	2017/01/20
3	Naomi	010-3333-5555	Hawaii	2017/01/10
4	Andra	010-6666-7777	Bora Bora	2017/01/11
5	Paul	010-9999-8888	Hawaii	2017/01/10
6	Colin	010-5555-4444	Venice	2017/01/16

다음 임무를 수행하기 위해 쿼리문을 작성하고 데이터베이스에 적용하십시오.

- BoraBora 여행은 공항테러 사태로 취소됐습니다. BoraBora 여행자의 명단을 삭제해주십시오.
- Hawaii 단체 관람객의 요청으로 출발일이 조정됐습니다. 2017/01/10에 출발하는 Hawaii의 출발일을 2017/01/17일로 수정해주십시오.
- 휴일 여행을 디즈니랜드로 떠나는 사람들의 핸드폰 번호를 조회하십시오.

▼ 풀이

```
# 데이터 입력 PUT customer POST customer/_doc/1 { "name": "Alfred", "phone": "010-1234-5678", "holiday_dest": "Disneyland", "departure_date": "2017/01/20" } POST customer/_doc/2 { "name": "Huey", "phone": "010-2222-4444", "holiday_dest": "Disneyland", "departure_date": "2017/01/20" } POST customer/_doc/3 { "name": "Naomi", "phone": "010-3333-5555", "holiday_dest": "Hawaii", "departure_date": "2017/01/10" } POST customer/_doc/4 { "name": "Andra", "phone": "010-6666-7777", "holiday_dest": "Bora Bora", "departure_date": "2017/01/11" } POST customer/_doc/5 { "name": "Paul", "phone": "010-9999-8888", "holiday_dest": "Hawaii", "departure_date": "2017/01/10" } POST customer/_doc/6 { "name": "Colin", "phone": "010-1234-5678", "holiday_dest": "Venice", "departure_date": "2017/01/16" } #1) BoraBora 여행은 공항테러 사태로 취소됐습니다. BoraBora 여행자의 명단을 삭제해주십시오. DELETE customer/_doc/4 #2) Hawaii 단체 관람객의 요청으로 출발일이 조정됐습니다. 2017/01/10에 출발하는 Hawaii의 출발일을 2017/01/17일로 수정해주십시오. POST customer/_update/3 { "doc": { "departure_date": "2017/01/17" } } POST customer/_update/5 { "doc": { "departure_date": "2017/01/17" } } #3) 휴일 여행을 디즈니랜드로 떠나는 사람들의 핸드폰 번호를 조회하십시오. GET customer/_doc/1 GET customer/_doc/2
```

배치프로세스

shakespeare.json 24733.9KB

accounts.zip 56.3KB

logs.jsonl.gz 8494.4KB

bulk API를 사용하면 다수의 작업을 한번에 진행할 수 있음

```
POST /customer/_bulk?pretty {"index":{"_id":"1"}} {"name": "John Doe" } {"inde
x":{"_id":"2"}} {"name": "Jane Doe" } [엔터]
```

bulk를 사용해 업데이트 및 삭제하는 예제

```
POST /customer/_bulk?pretty {"update":{"_id":"1"}} {"doc": { "name": "John Doe
becomes Jane Doe" } } {"delete":{"_id":"2"}} [엔터]
```

json 데이터를 사용해 다수의 작업을 한꺼번에 실행

배시

```
curl -H 'Content-Type: application/x-ndjson' -XPOST 'localhost:9200/bank/_bul
k?pretty' --data-binary @accounts.json
```

파워쉘

```
Invoke-RestMethod "http://localhost:9200/bank/_bulk" -Method Post -ContentType
'application/x-ndjson' -InFile "accounts.json"
```

검색 API

```
# bank 인덱스의 문서 검색 GET bank/_search?q=* GET bank/_search?q=Lynn GET bank/_s
earch?q=Pollard AND Lynn GET bank/_search?q=firstname:Lynn # bank 인덱스에서 특정
_source 필드만 검색 GET bank/_search?q=firstname:Lynn&_source=firstname,lastname
GET bank/_search?q=firstname:Lynn&_source=false # bank 인덱스에서 특정 필드로 정렬해서
검색 GET bank/_search?q=*&sort=balance GET bank/_search?q=*&sort=balance:desc #
원하는 위치에서 원하는 만큼의 데이터 질의 GET bank/_search?q=*&size=10&from=10 GET bank/
_search?q=*&size=10&from=20
```

검색 API 연습문제

TourCompany에입력했던 데이터가 모두 날아갔다. 이런 상황을 미리 방지하기 위해 벌크 데이터를 만들고 API를 사용하여 업로드 해보자. Index 이름은 tourcompany로 한다.

Doc Id	name	phone	holiday_dest	departure_date
1	Alfred	010-1234-5678	Disneyland	2017/01/20
2	Huey	010-2222-4444	Disneyland	2017/01/20
3	Naomi	010-3333-5555	Hawaii	2017/01/10
4	Andra	010-6666-7777	Borabora	2017/01/11
5	Paul	010-9999-8888	Hawaii	2017/01/10
6	Colin	010-5555-4444	Venice	2017/01/16

- 좀더 효과적인 임무 수행을 위해 검색 기능을 수행하는 쿼리를 작성하십시오.
  - tourcompany 인덱스에서 010-3333-5555를 검색하십시오.
  - 휴일 여행을 디즈니랜드로 떠나는 사람들의 핸드폰 번호를 조회하십시오(phone 필드만 출력).
  - departure date가 2017/01/10과 2017/01/11인 사람을 조회하고 이름 순으로 출력하십시오(name과 departure date 필드만 출력).
  - BoraBora 여행은 공항테러 사태로 취소됐습니다. BoraBora 여행자의 명단을 삭제해주십시오.
  - Hawaii 단체 관람객의 요청으로 출발일이 조정됐습니다. 2017/01/10에 출발하는 Hawaii의 출발일을 2017/01/17일로 수정해주십시오.

▼ 풀이

```
# 배치 프로세스 DELETE customer POST customer/_bulk {"index":{"_id":"1"}} {"name": "Alfred", "phone": "010-1234-5678", "holiday_dest": "Disneyland", "departure_date": "2017/01/20"} {"index":{"_id":"2"}} {"name": "Huey", "phone": "010-2222-4444", "holiday_dest": "Disneyland", "departure_date": "2017/01/20"} {"index":{"_id":"3"}} {"name": "Naomi", "phone": "010-3333-5555", "holiday_dest": "Hawaii", "departure_date": "2017/01/10"} {"index":{"_id":"4"}} {"name": "Andra", "phone": "010-6666-7777", "holiday_dest": "Bora Bora", "departure_date": "2017/01/11"} {"index":{"_id":"5"}} {"name": "Paul", "phone": "010-9999-8888", "holiday_dest": "Hawaii", "departure_date": "2017/01/10"} {"index":{"_id":"6"}} {"name": "Colin", "phone": "010-1234-5678", "holiday_dest": "Venice", "departure_date": "2017/01/16"} #1) tourcompany 인덱스에서 010-3333-5555를 검색하십시오. GET customer/_search?q="010-3333-5555" #2) 휴일 여행을 디즈니랜드로 떠나는 사람들의 핸드폰 번호를 조회하십시오(phone 필드만 출력). GET customer/_search?q=holiday_dest:Disneyland&_source=phone,holiday_dest #3) departure date가 2017/01/10과 2017/01/11인 사람을 조회하고 이름 순으로 출력하십시오.(name과 departure date 필드만 출력) POST customer/_search?q=departure_date:"2017/01/10" or departure_date:"2017/01/11"&_source=name,phone,holiday_dest&sort=name.keyword:asc #4) BoraBora 여행은 공항테러 사태로 취소됐습니다. BoraBora 여행자의 명단을 삭제해주십시오. POST customer/_update_by_query { "script": {"source": "ctx.op='delete'", "lang": "painless" }, "query": {"match": { "holiday_dest": "Bora Bora" } } } # 5) Hawaii 단체 관람객의 요청으로 출발일이 조정됐습니다. 2017/01/10에 출발하는 Hawaii의 출발일을 2017/01/17일로 수정해주십시오. POST customer/_update_by_query { "script": {"source": "ctx._source.departure_date='2017/01/17'", "lang": "painless"}, "query": { "bool": { "must": [ {"match": {"departure_date": "2017/01/10"}}, {"match": {"holiday_dest": "Hawaii"}} ] } } } GET customer/_search
```

대시보드 시각화를 위한 KIBANA 기초

매핑 작업

```
DELETE shakespeare DELETE logstash-2015.05.18 DELETE logstash-2015.05.19 DELETE logstash-2015.05.20 PUT shakespeare { "mappings" : { "properties" : { "speaker" : {"type": "keyword" }, "play_name" : {"type": "keyword" }, "line_id" : { "type" : "integer" }, "speech_number" : { "type" : "integer" } } } } #18~20까지 세개의 인덱스 구성 필요 PUT logstash-2015.05.18 { "mappings": { "properties": { "geo": { "properties": { "coordinates": { "type": "geo_point" } } } } } } PUT logstash-2015.05.19 { "mappings": { "properties": { "geo": { "properties": { "coordinates": { "type": "geo_point" } } } } } } PUT /logstash-2015.05.20 { "mappings": { "properties": { "geo": { "properties": { "coordinates": { "type": "geo_point" } } } } } }
```

배치프로세스를 사용해 업로드 진행(배시)

```
curl -H 'Content-Type: application/x-ndjson' -XPOST 'localhost:9200/bank/_bulk?pretty' --data-binary @accounts.json curl -H 'Content-Type: application/x-ndjson' -XPOST 'localhost:9200/_bulk?pretty' --data-binary @shakespeare.json curl -H 'Content-Type: application/x-ndjson' -XPOST 'localhost:9200/_bulk?pretty' --data-binary @logs.jsonl
```

배치프로세스를 사용해 업로드 진행(파워쉘)

```
Invoke-RestMethod "http://localhost:9200/bank/_bulk" -Method Post -ContentType 'application/x-ndjson' -InFile "accounts.json" Invoke-RestMethod "http://localhost:9200/_bulk" -Method Post -ContentType 'application/x-ndjson' -InFile "shakespeare.json" Invoke-RestMethod "http://localhost:9200/_bulk" -Method Post -ContentType 'application/x-ndjson' -InFile "logs.jsonl"
```

시각화는 교재를 참고

파일 비트를 활용한 아파치 서버 로그 수집

쿠버네티스 실습: ubuntu-20.04.7z 다운로드 - VMware

https://drive.google.com/file/d/1xy4_N4xmLGQMHt59ZMMJjsTA62oYOA7a/view?usp=sharing

id: user01

pw: test1234

아파치 설치하기

```
apt install apache2 -y
```

파일비트 7.14.1 deb 32비트 다운로드 명령어 실행

```
wget https://artifacts.elastic.co/downloads/beats/filebeat/filebeat-7.14.1-amd64.deb --no-check-certificate sudo dpkg -i filebeat-7.14.1-amd64.deb sudo vim /etc/filebeat/filebeat.yml
```

output.elasticsearch를 찾아서 호스트 위치가 정확한지 확인

파일비트 모듈 설정으로 apache2 로그 수집

```
cd /etc/filebeat/modules.d cp apache.yml.disabled apache.yml vim apache.yml
```

모듈에 다음과 같이 경로를 설정

```
- module: apache access: enabled: true var.paths: ["/var/log/apache2/access.log*"] error: enabled: true var.paths: ["/var/log/apache2/error.log*"]
```

설정을 위해 재시작

```
sudo /etc/init.d/filebeat restart journalctl -u filebeat # 로그 확인
```

인덱스 추가 후 Suricata 필드에서 파싱된 로그 확인하기