



클러스터 운영

오늘의 학습내용

- 무중단 운영을 위한 롤링 리스타트
- 안정적인 성능 제공을 위한 샤드 분배 방법
- index setting
- 미리 정의된 template으로 인덱싱하기

◆ 무중단 운영을 위한 롤링 리스타트

■ 롤링 리스타트

- ❖ 무중단 운영을 위한 작업 방법
- ❖ 시스템 작업이나 Elasticsearch version upgrade 를 해야 하는 상황
- ❖ replica가 있는 cluster의 경우 application restart나 system reboot을 할 때마다 cluster 내부에서 할당되지 않은 샤드들(unassigned)이 기본 라우팅 설정에 따라 복구를 위해 자동으로 재분배
- ❖ 많은 노드들을 작업해야 할 때 이런 shard들이 재분배 되기를 기다렸다가 cluster가 green상태가 될 때 까지는 시간 뿐 아니라 네트워크와 Disk I/O 등의 많은 리소스를 필요
- ❖ 위와 같은 작업을 할 때 리밸런싱이 일어나지 않게 하는 것이 Rolling Restart
- ❖ `_cluster/settings`의 `cluster.routing.allocation.enable` 값 변경에 따른 설정 변경
 - shard 종류에 따라 어떻게 처리 할 것인지 결정: none, all

◆ 무중단 운영을 위한 롤링 리스타트

■ 롤링 리스타트 이슈

- ❖ 패치 및 업그레이드(OS 및 애플리케이션) 시 이슈사항 예방
- ❖ 한 대씩 바로 업데이트 및 리스타트 시 "전체 클러스터"의 샤드 재분배 작업이 발생
- ❖ 노드 재실행 시 2번의 클러스터 샤드 재분배(stop 및 start시)로 인한 I/O 부하가 발생
- ❖ shard allocation 옵션 disable 이후 shutdown (해당 노드 kill 시, 재 배치 작업 발생하지 않음)
- ❖ 엘라스틱서치 실행 및 allocation 옵션 all 로 처리(all로 처리시 unassigned 된 샤드가 노드에 할당)

▲ 안정적인 성능 제공을 위한 샤드 분배 방법

■ Elasticsearch 데이터 구조

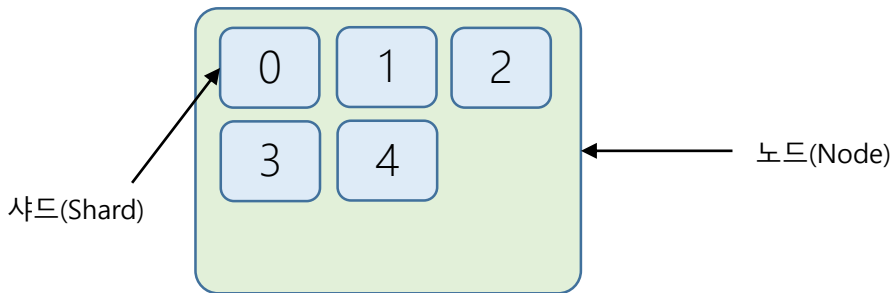
- ❖ 엘라스틱서치는 데이터(document)를 엘라스틱서치 인덱스로 만든 뒤 샤드로 분리하여 보관
- ❖ 샤드는 논리적/물리적으로 분할된 인덱스이며, 각각의 엘라스틱서치 샤드는 루씬 인덱스
- ❖ 루씬은 새로운 데이터(document)를 엘라스틱서치 인덱스에 저장할 때 "세그먼트(segment)"를 생성

Index							
Shard		Shard		Shard		Shard	
Lucene Index		Lucene Index		Lucene Index		Lucene Index	
Segment	Segment	Segment	Segment	Segment	Segment	Segment	Segment

▲ 안정적인 성능 제공을 위한 샤드 분배 방법

■ Index & Shards

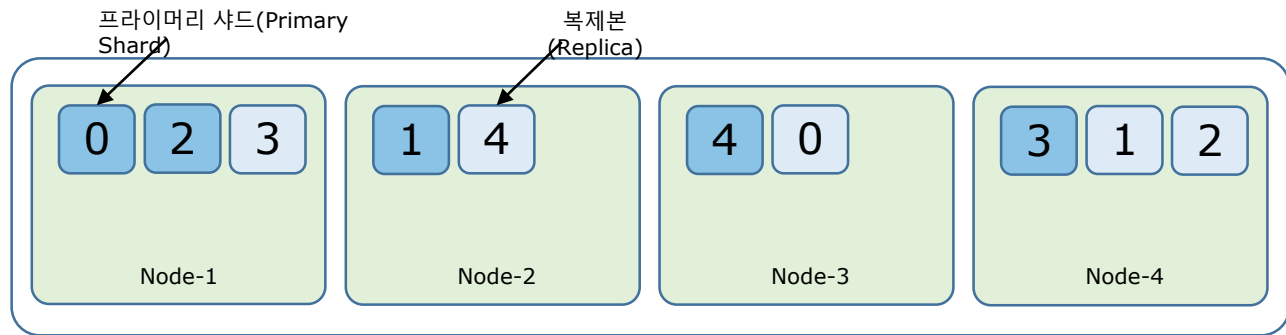
- ❖ 도큐먼트(document) : Elasticsearch 에서 단일 데이터 단위
- ❖ 인덱스(Index) : 도큐먼트를 모아놓은 집합
- ❖ 인덱스는 기본적으로 샤드(shard)라는 단위로 분리되고 각 노드에 분산되어 저장
- ❖ 샤드는 루씬의 단일 검색 인스턴스
- ❖ 하나의 인덱스가 5개의 샤드로 저장 예



▲ 안정적인 성능 제공을 위한 샤드 분배 방법

■ 프라이머리 샤드(Primary Shard)와 복제본(Replica)

- ❖ 클러스터에 노드를 추가하게 되면 샤드들이 각 노드들로 분산되고 디폴트로 1개의 복제본을 생성(7.x)
- ❖ 처음 생성된 샤드를 프라이머리 샤드(Primary Shard), 복제본은 리플리카(Replica)
- ❖ 같은 샤드와 복제본은 동일한 데이터를 담고 있으며 반드시 서로 다른 노드에 저장
- ❖ 한 인덱스가 5개의 샤드로 구성되어 있고, 클러스터가 4개의 노드로 구성 예



▲ 안정적인 성능 제공을 위한 샤드 분배 방법

■ shard 종류에 따라 어떻게 처리 할 것인지 결정

- ❖ Elasticsearch 운영 중 여러 노드들로 구성된 클러스터에 노드별 용량이 상이해지는 경우가 발생
- ❖ 생성되는 인덱스의 샤드가 노드수와 동일하다면 큰 차이가 발생하지 않음
- ❖ 운영 중 노드 증설 등으로 인해 기존에 계획한대로 샤드 배치가 되지 않을 수 있음
- ❖ 노드에 샤드가 똑같이 분배되지 않을 때에 용량 격차가 벌어짐
- ❖ 방법
 - POST _cluster/reroute 사용
 - PUT _cluster/settings의 disk threshold 사용
 - 샤드 강제 분배

◆ index setting

■ 인덱스 설정

❖ Static index settings

- 인덱스가 생성될 때 설정, 일부는 closed index에서 설정 가능
- number_of_shards: 샤드의 개수

❖ Dynamic index settings

- live index에서 update-index-settings API를 사용해 변경 가능
- 운영 중 인덱스 세팅 변경
- RestAPI로 변경사항 요청
- number_of_replicas : 운영 중에 리플리카 샤드 개수를 변경
- refresh_interval : 세그먼트에 저장된 데이터를 검색할 수 있도록 commit point를 생성하는 주기
- routing.allocation.enable : 데이터 노드에 샤드를 어떤 방식으로 할당할 것인지를 결정
- routing.rebalance.enable : 데이터 노드에 샤드를 어떤 방식으로 재배포할 것인지를 결정

◆ index setting

■ Mapping

- ❖ document가 indexing 될 때, document와 document에 포함된 field들을 어떻게 저장할지를 결정하는 과정
- ❖ 6.x 부터 Multi Mapping Deprecated
- ❖ Dynamic Mapping
 - Elasticsearch가 인입되는 document를 보고, 알아서 타입을 찾아 mapping
- ❖ Static mapping
 - 사용자가 정의한 스키마를 기준으로 mapping

▲ 미리 정의된 Template으로 인덱싱하기

■ Elasticsearch Index templates

- ❖ Index template은 새로운 index가 생성될 때에, index 설정 또는 특정 필드의 데이터 타입을 정의
- ❖ Elasticsearch에 Logstash 등 수집기로 데이터 수집 전 index가 사전 생성될 필요가 없음
- ❖ index template를 생성함으로써 index별로 Shard 개수나 특정 필드에 대한 개별적인 관리 가능

▲ 미리 정의된 Template으로 인덱싱하기

■ Template 인덱싱하기

- ❖ 미리 정의된 템플릿으로 인덱싱하기
- ❖ 인덱스가 생성될 때 사용자 정의된 세팅이나 매핑을 자동으로 적용
- ❖ 인덱스 패턴, 인덱스 세팅, 인덱스 매핑 관련 사항 정의
- ❖ 인덱스가 생성될 때 패턴이 매칭되는 인덱스는 해당 정의를 따름
- ❖ order가 높은 번호가 낮은 번호를 override하여 merging

◆ 미리 정의된 Template으로 인덱싱하기

■ 템플릿 생성하기

```
$ curl -XPUT localhost:9200/_template/httpd-access-log -d `
{
  "index_patterns": [
    "httpd-access-*"
  ],
  "mappings": {
    "log": { // type name
      "properties": {
        "ip": {
          "type": "text"
        },
        "host": {
          "type": "keyword"
        },
        "uri": {
          "type": "text"
        },
        "datetime": {
          "type": "date"
        },
        "@timestamp": {
          "type": "date" }...
      }
    }
  }
}
```

◆ 미리 정의된 Template으로 인덱싱하기

■ Index setting 설정하기

```
$ curl -XPUT localhost:9200/_template/httpd-access-log -d '{
  "index_patterns": [
    "httpd-access-*"
  ],
  "settings": { // setting
    "number_of_shards": 1
  },
  "mappings": {
    "log": { // type name
      "properties": {
        "ip": {
          "type": "text"
        },
        "host": {
          "type": "keyword"
        },
        "uri": {
          "type": "text"
        },
        ...
      }
    }
  }
}
```