Elasticsearch 환경 설정 및 클러스터 운영(3~4회차)

목차

- Elasticsearch 환경설정
- Elasticsearch 클러스터 운영
- Elasticsearch API 활용하기
- Q & A

Elasticsearch 환경설정

elasticsearch.yml - Elasticsearch 의 핵심 설정

jvm.options - JVM 옵션, 힙사이즈의 중요성

log4j2.properties - 로그는 어떻게 모을 것인가

Elasticsearch 환경설정

Elasticsearch 환경설정

1) Static settings
- elasticsearch.yml 파일에 정의
- 노드 별로 설정파일에 설정

2) Dynamic Settings
- 클러스터에 API 로 호출
- 클러스터 단위로 설정

Elasticsearch 환경설정 - elasticsearch.yml

elasticsearch.yml 파일 환경설정

cluster.name
- 클러스터를 고유하게 식별할 수 있는 이름 설정

ex) cluster.name: tuto-es

node.name
- 노드를 고유하게 식별할 수 있는 이름 설정
- 호스트명 기준으로 설정하는 것이 운영에 용이

ex) node.name: tuto-es-data01

- 7.x 버전부터는 기본값이 호스트명으로 설정됨

Elasticsearch 환경설정 - elasticsearch.yml

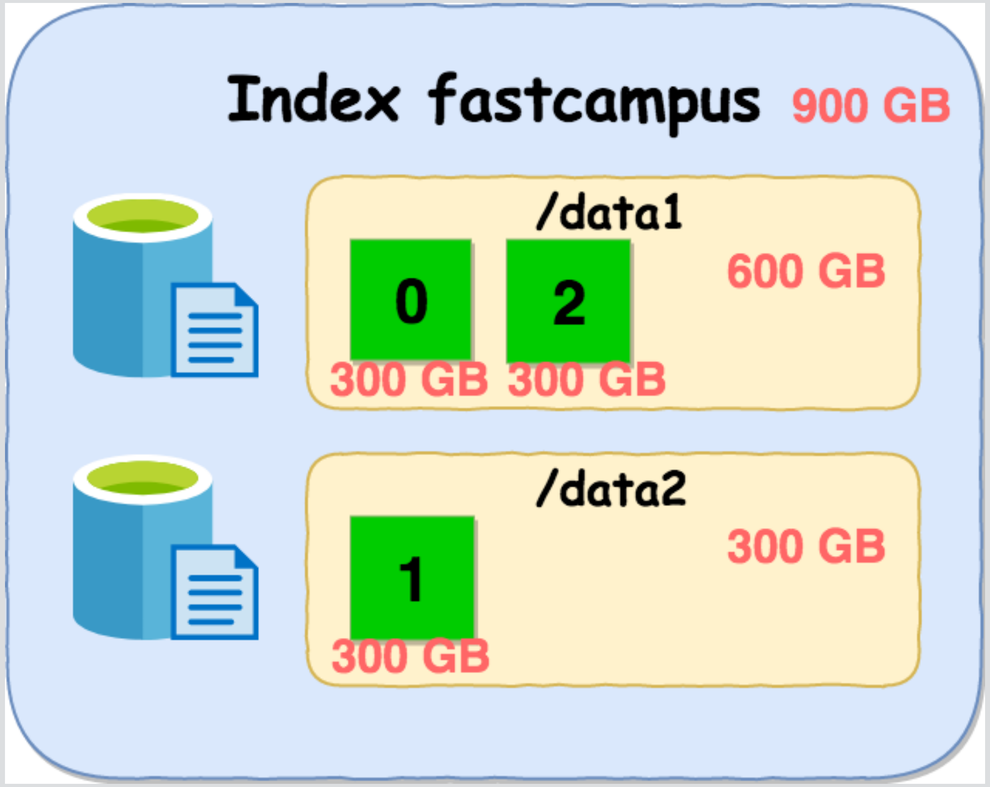
path.data
- Index 데이터를 저장할 경로 지정
- Single Path 혹은 Multi Path 사용 가능

ex) path.data: /data1
path.data: /data1, /data2

- Multi Path 를 쓸 경우 샤드 계획이 잘 수립되어야..

path.logs
- Elasticsearch 의 로그를 저장할 경로 지정
- 어플리케이션 운영 로그, Elasticsearch Deprecated 로그, Indexing, Searching Slow 로그

ex) path.logs: /var/log/elasticsearch



Elasticsearch 환경설정 - elasticsearch.yml

Discovery
- 노드가 클러스터를 찾아가는 과정
- Ping 을 기반으로 동작(default interval 1s, retries 3)

discovery.seed_hosts
- 기본 설치 진행 시 localhost 내에 9300-9305 포트를 스캔하여 클러스터링 진행
- 외부 시스템에 설치된 ES 노드와 클러스터링이 필요하면 필수로 설정
- Data 노드에는 이 설정만으로 Discovery

-> 6.x 에서는 discovery.zen.ping.unicast.hosts 로 정의

ex) discovery.seed_hosts: ["1.1.1.1:9300", "1.1.1.2:9300", "2.2.2.1:9300",]

cluster.initial_master_nodes
- 최초 구성 시 마스터 선출 가능 목록을 구성하는 설정
- 단일 노드의 경우에도 단일 노드의 아이피 필수로 설정
- 목록을 기준으로 minimum master nodes 수를 계산/적용
- Master 노드에서는 이 설정과 discovery.seed_hosts 설정 으로 Discovery

-> 6.x 에서는 discovery.zen.minimum_master_nodes 로 정의

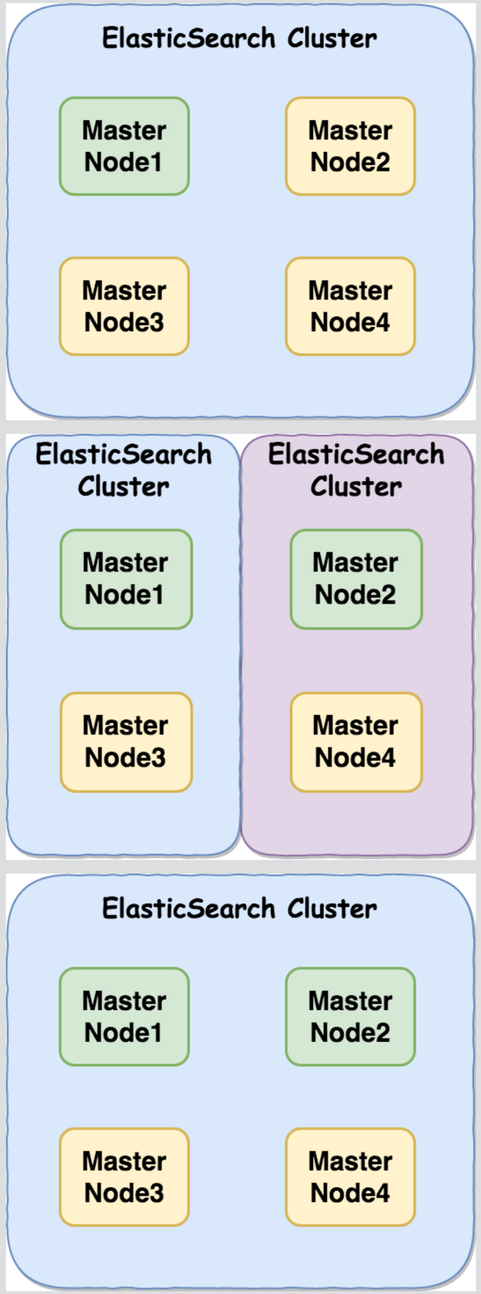
ex) cluster.initial_master_nodes: ["1.1.1.1:9300", "1.1.1.2:9300", "2.2.2.1:9300",]

Elasticsearch 환경설정 - elasticsearch.yml

discovery.zen.minimum_master_nodes
- 최소 마스터 개수 설정
- 마스터노드 개수 / 2 + 1 개 설정
- 해당 노드 개수만큼 마스터가 내려가면
  데이터 무결성을 위해 클러스터 중지

Split Brain 이란?
- 클러스터 구성에서 네트워크 단절로 인해
  여러개의 노드가 서로 마스터로 인식되는 증상
- 4개의 마스터를 운영할 때에는 최소 마스터 개수를
  4 / 2 + 1 = 3 으로 설정
- 2대가 내려가는 순간 클러스터를 중지시켜 Split Brain 을 방지

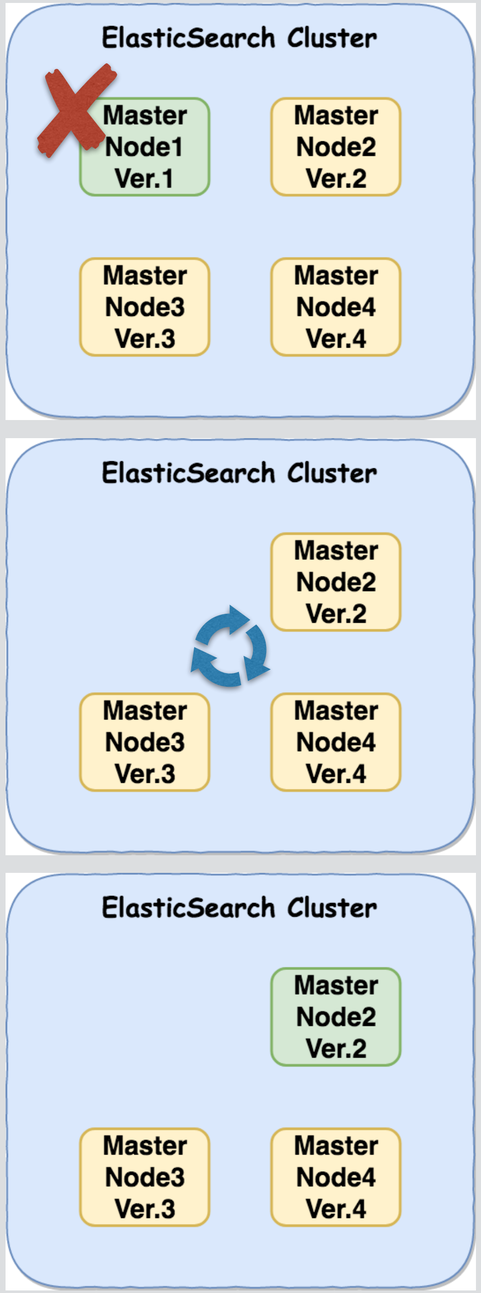
7.x 에서는 자동으로 최소 마스터 개수를 설정



Elasticsearch 환경설정 - elasticsearch.yml

마스터 Fault

- 마스터로 정의된 노드들은 각각 cluster state version 을 갖음
- 실제 마스터가 내려가면 각각의 마스터 노드들은 zen discovery 에 정의된 호스트에게 Ping Check 를 시작
- 응답이 오는 호스트 중 cluster state version 이 가장 낮은 호스트를 마스터로 선출



Elasticsearch 환경설정 - elasticsearch.yml

Network 설정

network.host 고급 설정

network.bind_host
- network.host 설정에서 외부의 데이터 호출을 받는 부분만 분리
- 여러개의 network device 에 설정된 ip 다중으로 설정 가능

ex) network.bind_host: 0.0.0.0

network.publish_host
- 클러스터 내의 다른 노드들과 통신을 하는 부분만 분리
- 노드를 고유하게 식별할 수 있는 하나의 ip 만 설정 가능

ex) network.publish_host: 10.190.5.5

Elasticsearch 환경설정 - elasticsearch.yml

Network 설정

http.port
- http 프로토콜을 통해 Elasticsearch 의 API 를 전달할 때 사용할 포트 설정

ex) http.port: 9200

transport.tcp.port
- 클러스터 내에 노드들이 서로 통신을 할 때 사용할 포트 설정
- 노드는 서로의 용량이나 샤드의 상태를 알아야하기 때문에 tcp 통신을 해야함

ex) transport.tcp.port: 9300

Elasticsearch 환경설정 - elasticsearch.yml

Node Roles 설정
- 노드의 role 에는 master-eligible, data, ingest, coordinate role 가 있음

Master-eligible Node
- 마스터 노드로서의 역할을 할 수 있는 role 이 부여된 노드

```
node.master: true
node.data: false
node.ingest: false
```

Data Node
- 데이터가 저장되는 역할을 할 수 있는 role 이 부여된 노드

```
master.node: false
node.data: true
node.ingest: false
```

Elasticsearch 환경설정 - elasticsearch.yml

Ingest Node
- 문서가 인덱싱 되기 전에 파이프라인을 통해 사전처리를 할 수 있는 role 이 부여된 노드
- 기본값은 true, 보통 client node 세팅으로 사용

```
node.master: false
node.data: false
node.ingest: true
```

Client Node
- 클라이언트의 요청을 받고 라우팅 및 분산만 처리할 수 있는 role 이 부여된 노드

```
master.node: false
node.data: false
node.ingest: false
```

Elasticsearch 환경설정 - elasticsearch.yml

그 외 설정..

http.cors.enabled: true
- 웹 브라우저에서 Elasticsearch 에 접근할 수 있도록 해주는 설정
- Head 나 HQ 플러그인을 사용할 때 설정

http.cors.allow-origin: "*"
- 웹 브라우저로 접근할 수 있는 IP ACL 설정

실습

Elasticsearch Clustering
https://github.com/benjamin-btn/ES7-Tutorial/tree/master/ES-Tutorial-3-1

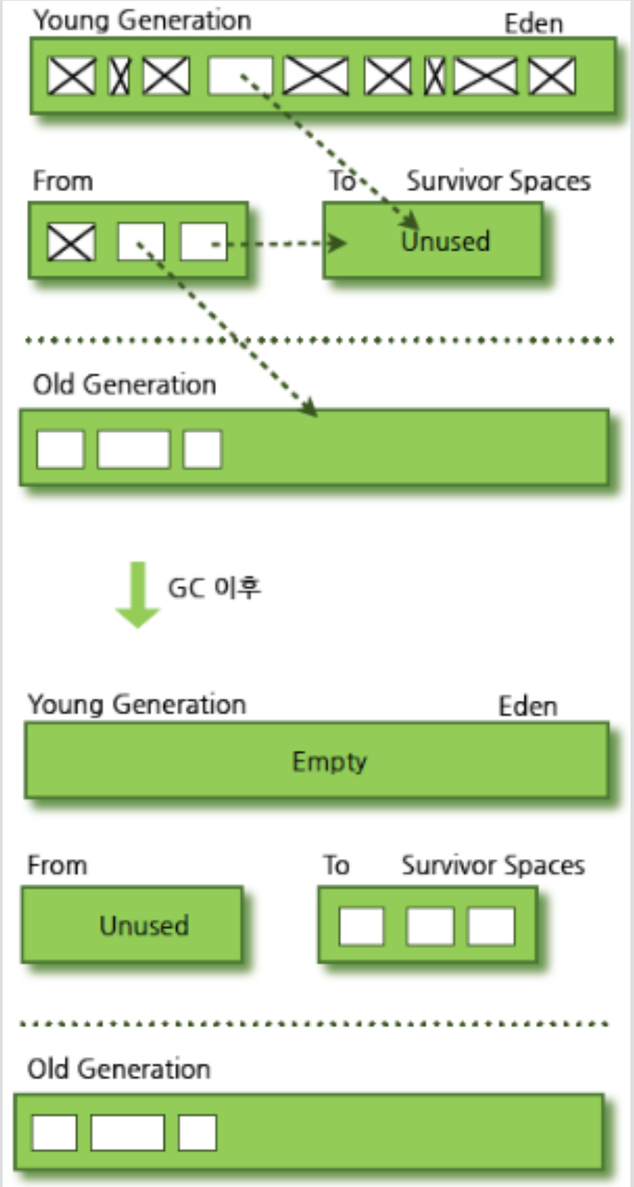
Elasticsearch 환경설정 - jvm.options

Java 는 Heap 에 객체를 할당하여 사용하는 구조
Heap 영역은 크게 Young / Old Generation 으로 구성
알아서 필요없는 객체를 청소해주는 GC(Garbage Collector)

Young Generation
- 가장 처음에 객체를 받는 Eden 영역
- Eden 이 차올라 age bit 에 의해 정리되는 객체를 제외한 나머지 객체를 넘겨주는 from Survivor 영역
- 다시 Eden 이 차오를 때 from Survivor 영역의 객체와 함께 객체를 넘겨주는 to Survivor 영역

Old Generation
- from Survivor 영역이 차올라 to Survivor 영역이 받아줄 수 없을 때 객체를 받아주는 영역
- age bit 에 의해 정리되지 않은 객체를 받아주는 영역

GC 는 위 영역들 사이에서 객체가 이동될 때 발생
GC 가 진행될 때에는 모든 객체 할당 행위를 멈추고 객체의 복사만 진행되는 Stop The World 발생



Elasticsearch 환경설정 - jvm.options

-Xms4g
- 최소 힙사이즈 크기 설정

-Xmx4g
- 최대 힙사이즈 크기 설정

initial size 와 maximum size 를 동일하게 설정 권고
- runtime 에서 힙 사이즈 조정 비용이 큼
- heap size 조정 중 JVM 이 잠시 멈출 수 있음 (Stop the world)

크면 클수록 많은 데이터를 Heap 에서 사용 가능
- GC 발생 시 성능 저하 고려

Elasticsearch 환경설정 - jvm.options

가능하면 물리 메모리의 50% 를 넘지 않도록 권고

- 최초 인덱싱이 일어날 때 시스템 버퍼 캐시를 통해 segment 로 적재

- 검색이나 어그리게이션이 일어날 때 버퍼 캐시에 있는 segment 를 확인

- 디스크 I/O 를 피하기 위한 충분한 버퍼 캐시를 확보하도록 권고

32G 를 넘지 않도록 권고

- Heap 에 데이터를 OOP(Ordinary Object Point)
  라는 구조체로 저장

- 아키텍쳐 별로 32bit 와 64bit 크기의 주소 참조

- 32bit 는 최대 32G 까지 참조 가능(offset 영역을 활용), 64bit 는 18EB 까지 참조 가능

- 64bit 는 메모리 참조의 영역이 넓어 성능 저하

Elasticsearch 환경설정 - jvm.options

-XX:+UseConcMarkSweepGC
- 기본으로 CMS GC 를 사용

-XX:CMSInitiatingOccupancyFraction=75
- Old 영역이 75% 차오르면 GC 주기를 시작

-XX:+UseCMSInitiatingOccupancyOnly
- GC 통계에 따르지 않고 설정한 CMSInitiatingOccupancyFraction 을 기준으로 GC 주기를 시작

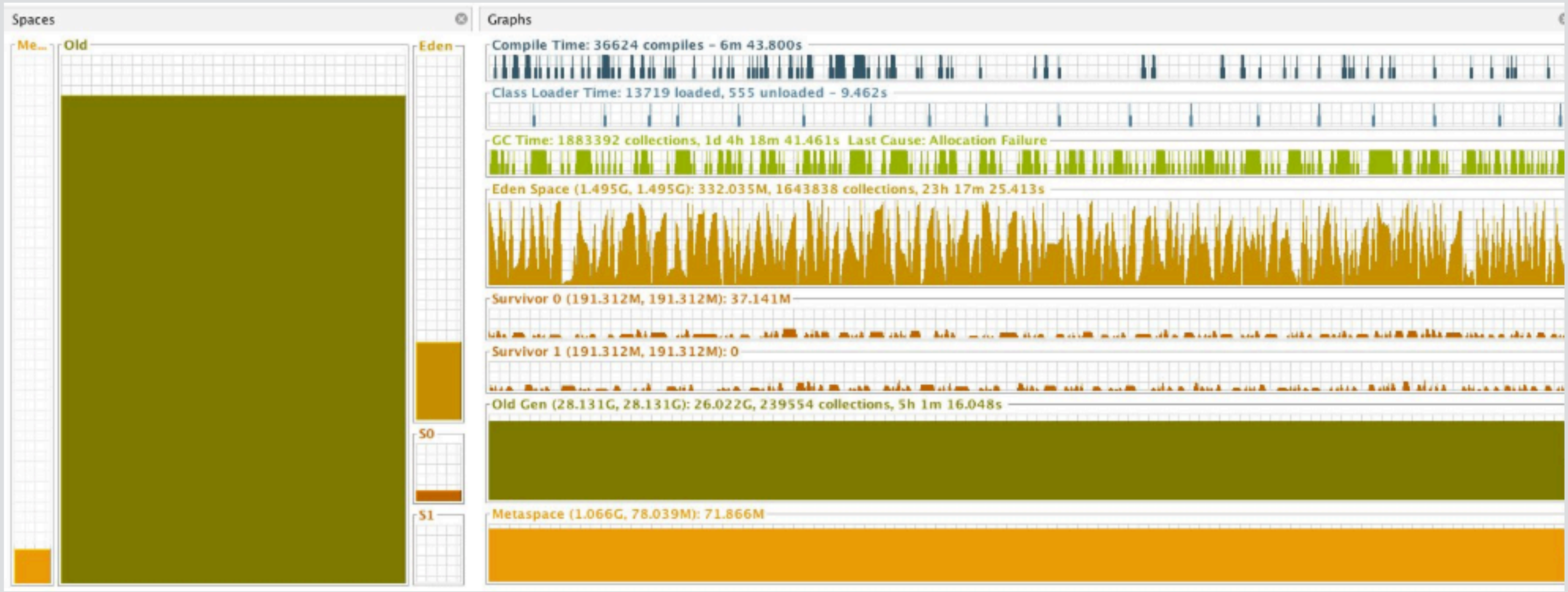
-XX:+HeapDumpOnOutOfMemoryError
- OOM 에러 발생 시 힙덤프를 발생시켜줌

-XX:HeapDumpPath=/var/lib/elasticsearch
- 힙 덤프를 저장할 경로

-XX:ErrorFile=/var/log/elasticsearch/hs_err_pid%p.log
- JVM Fatal error logs 를 받을 경로

Elasticsearch 환경설정 - jvm.options

CMS 기본 GC 의 JVM Heap Space



Elasticsearch 환경설정 - jvm.options

GC Tuning

Young 영역을 구성하는 Eden, Survivor 0, 1 이 작아 잦은 young gc 발생

age bit 도달 전에 old gc 에 의해 객체가 old 영역으로 이동

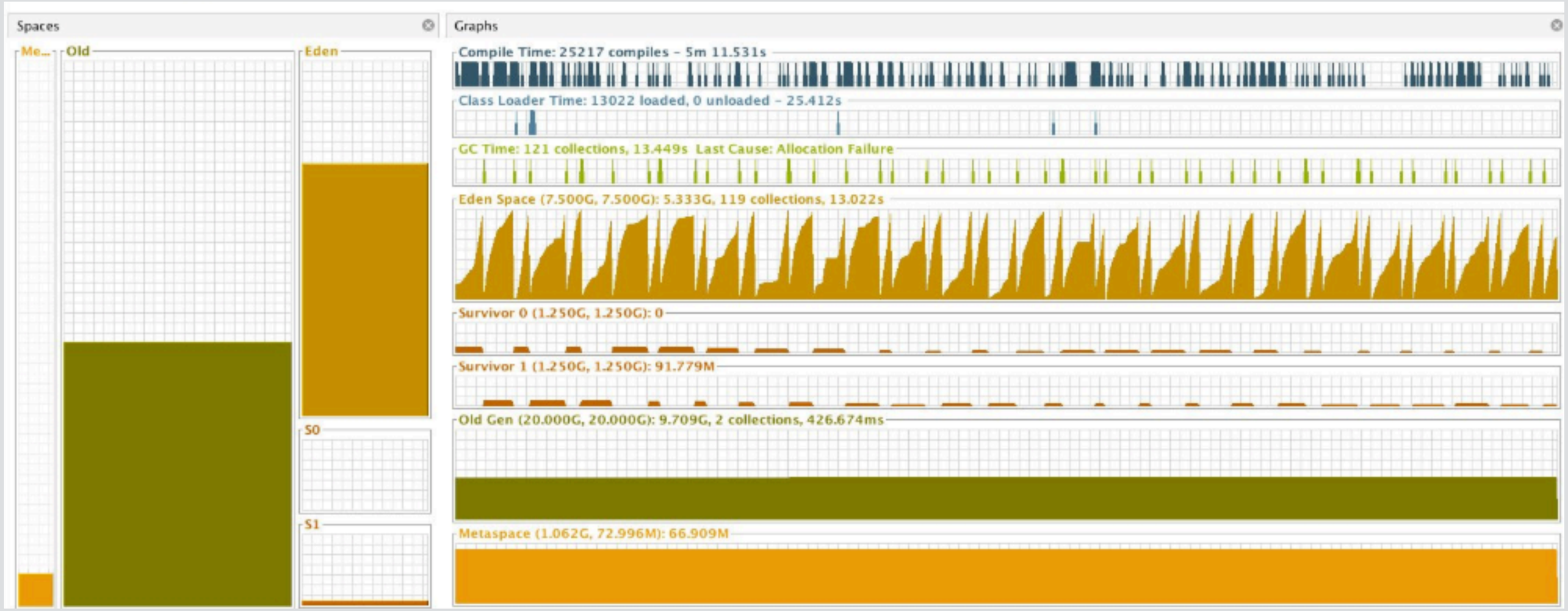
Young 영역을 좀 더 확보하여 young gc 의 빈도를 줄이는 게 성능 확보에 유리

-XX:NewRatio=2
-> New:Old = 1:2 로 튜닝

-XX:SurvivorRatio=6
-> Survivor0:Survivor1:Eden = 1:1:6 로 튜닝

Elasticsearch 환경설정 - jvm.options

CMS 튜닝된 GC 의 JVM Heap Space



Elasticsearch 환경설정 - log4j2.properties

Elasticsearch 는 log4j 2 를 사용해 어플리케이션 로그를 기록

${sys:es.logs.base_path}
- Log 설정 디렉토리
- path.logs

${sys:es.logs.cluster_name}
- 클러스터 이름
- cluster.name

${sys:es.logs.node_name}
- 노드 이름
- node.name

ex) ${sys:es.logs.base_path}${sys:file.separator}${sys:es.logs.cluster_name}.log
/var/log/elasticsearch/mycluster.log

Elasticsearch 환경설정 - log4j2.properties

로그의 종류

${sys:es.logs.base_path}${sys:file.separator}${sys:es.logs.cluster_name}.log
- 클러스터 운영로그 설정

${sys:es.logs.base_path}${sys:file.separator}${sys:es.logs.cluster_name}_deprecation.log
- Elasticsearch 에서 수행되고 있는 Deprecated 된 기능 정보

${sys:es.logs.base_path}${sys:file.separator}${sys:es.logs.cluster_name}_index_search_slowlog.log
- 인덱스 검색 슬로우 로그 정보

${sys:es.logs.base_path}${sys:file.separator}${sys:es.logs.cluster_name}_index_indexing_slowlog.log
- 인덱스 인덱싱 슬로우 로그 정보

${sys:es.logs.base_path}${sys:file.separator}${sys:es.logs.cluster_name}_access.log
- X-Pack auditing 로그 정보

Elasticsearch 환경설정 - 그 외 시스템 설정

Elasticsearch 는 많은 파일에 다량의 접근 시도
열 수 있는 File descriptor 가 부족하면 데이터 손실 가능성 발생
sudo vi /etc/security/limits.conf

```
elasticsearch              soft    nofile          65536
elasticsearch              hard    nofile          65536
```

Elasticsearch 는 operations type 에 따라 많은 thread pool 을 사용
elasticsearch 유저가 적어도 4096 개의 프로세스를 다룰 수 있어야 함

```
elasticsearch              soft    noproc          4096
elasticsearch              hard    noproc          4096
```

Elasticsearch 환경 변수가 정의되어 있는 파일
sudo vi /etc/sysconfig/elasticsearch

Elasticsearch 환경설정 - 그 외 시스템 설정

Elasticsearch 는 인덱스를 Filesystem 에 쓸 때 mmap 을 사용

```
$ sudo vi /etc/sysctl.conf
vm.max_map_count=262144

$ sudo sysctl -p
```

Swap Disabling

- 디스크로 swap out 이 되면 성능 저하가 발생

```
$ sudo swapoff -a

$ sudo vi /etc/sysctl.conf
vm.swappiness = 1

$ sudo sysctl -p
```

Elasticsearch 환경설정 - 그 외 시스템 설정

More Settings..
https://www.elastic.co/guide/en/elasticsearch/reference/current/important-settings.html
https://www.elastic.co/guide/en/elasticsearch/reference/current/system-config.html

실습

Elasticsearch Master/Data node
https://github.com/benjamin-btn/ES7-Tutorial/tree/master/ES-Tutorial-3-2

Elasticsearch 클러스터 운영

Rolling Restart - 무중단 운영을 위한 작업방법

Shard Allocation - 안정적인 성능 제공을 위한 샤드 분배 방법

Index setting - 인덱스는 어떻게 설정하여야 할까

Template - 미리 정의된 템플릿으로 인덱싱하기

hot-warm data node - 비용 절감하기

Elasticsearch 클러스터 운영 - Rolling Restart

Rolling Restart

시스템 작업이나 Elasticsearch Version upgrade 를 해야하는 니즈

리플리카가 있는 클러스터의 경우, 어플리케이션 재시작이나 시스템 리붓을 할 때마다
클러스터 내부에서는 주인을 잃은 샤드들이 기본 라우팅 설정에 의해 복구를 위해 자동으로 재할당

많은 노드들을 작업해야 할 때면 이런 샤드들이 재할당 되기를 기다렸다가
클러스터가 green 상태가 될 때 까지는 시간 뿐 아니라 네트워크와 Disk I/O 등의 많은 리소스를 필요

이런 작업을 할 때에 샤드의 재할당이 일어나지 않게 하는 것이 Rolling Restart

Elasticsearch 클러스터 운영 - Rolling Restart

_cluster API 로 클러스터 라우팅 할당을 new_primaries 로 변경

Primary / Replica 샤드 간 데이터 동기화

```
PUT _cluster/settings
{
   "transient" : {
      "cluster.routing.allocation.enable" : “new_primaries”
   }
}

POST _flush/synced
```

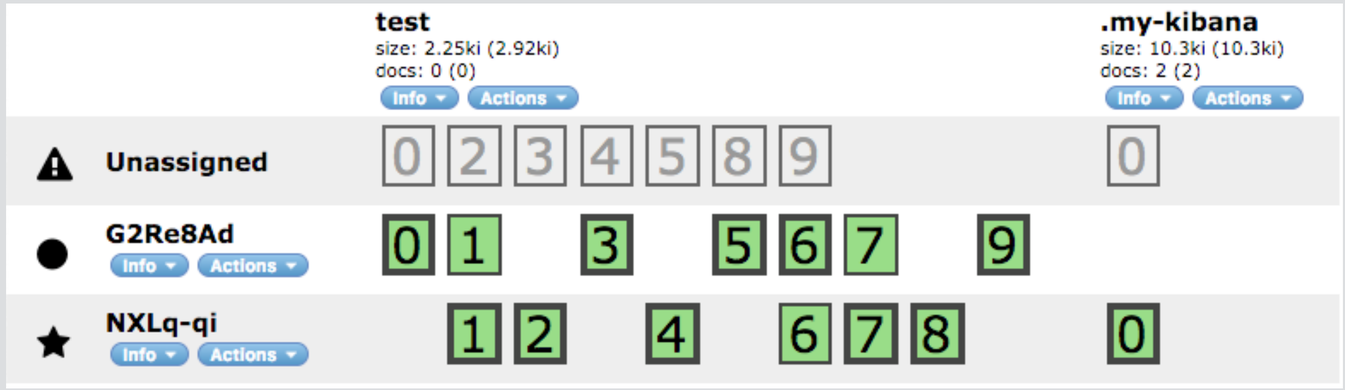
Elasticsearch 클러스터 운영 - Rolling Restart

작업하고자 하는 노드의 프로세스 중지

$ sudo systemctl stop elasticsearch

이후, 클러스터에서 제외된 노드 내의 샤드들이
unassigned 상태로 변경됨

클러스터 상태는 yellow 로 전환

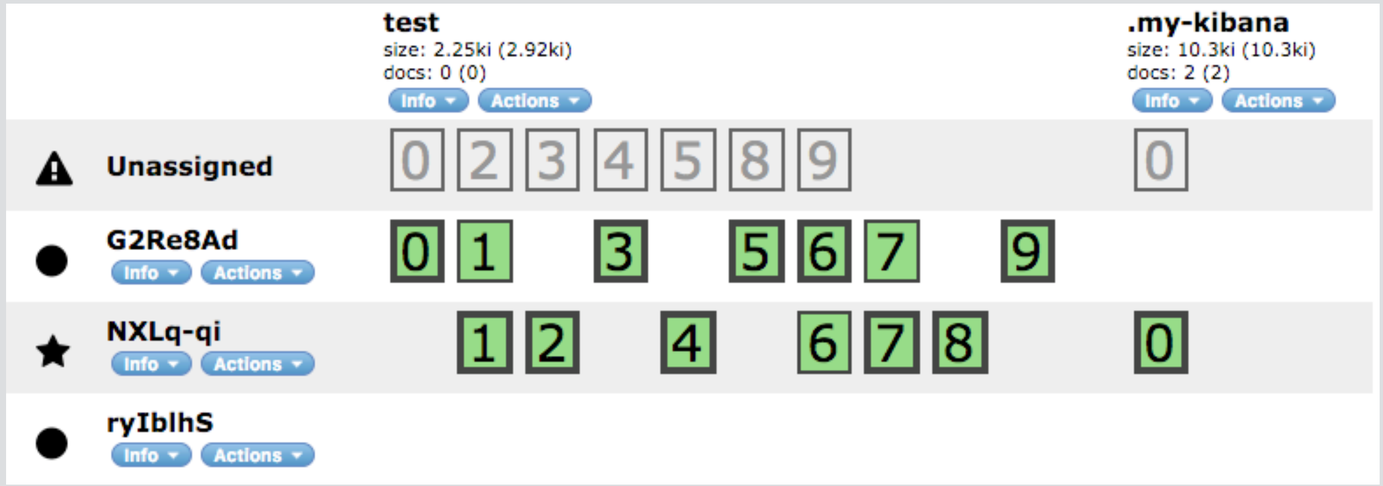


Elasticsearch 클러스터 운영 - Rolling Restart

작업 진행 후, ES 프로세스를 재시작

$ sudo systemctl start elasticsearch

프로세스가 제대로 올라오면 클러스터에
해당 노드가 샤드가 없는 상태로 추가됨



Elasticsearch 클러스터 운영 - Rolling Restart

클러스터에 추가된 것을 확인했으면 라우팅 할당 on

```
PUT _cluster/settings
{
   "transient" : {
      "cluster.routing.allocation.enable" : null
   }
}
```

이후에 unassigned 샤드들이 올라온 노드로 복구
위의 과정을 노드별로 반복

Elasticsearch 클러스터 운영 - Rolling Restart

Rolling Restart More..
https://www.elastic.co/guide/en/elasticsearch/reference/current/rolling-upgrades.html

Elasticsearch 클러스터 운영 - Shard Allocation

Shard Allocation

ES 를 운영하다보면 여러대로 구성된 클러스터에 노드별 용량이 상이해지는 경우 발생

생성되는 인덱스의 샤드가 노드수와 동일하다면 큰 차이가 발생하지 않겠지만
운영하다보면 노드 증설 등으로 인해 기존에 계획한대로 샤드 배치가 되지 않음

노드에 샤드가 똑같이 분배되지 않을 때에 용량 이격은 벌어짐

노드 간 볼륨사용량 이격이 벌어지는 이유

생성되는 인덱스의 샤드 개수가 노드 개수와 다를 때
- 샤드 개수를 계획할 때 노드 개수를 고려해야 함

Elasticsearch 클러스터 운영 - Shard Allocation

POST _cluster/reroute 를 이용한 샤드 강제 분배
이 방법은 클러스터의 reroute 를 이용

```
POST _cluster/reroute
{
   "commands" : [ {
      "move" : {
         "index" : "twitter",
         "shard" : 0,
         "from_node" : "G2Re8Ad",
         "to_node" : "ryIblhS"
      }
   } ]
}
```


```
{
  "acknowledged": true,
  "state": {
    "cluster_uuid": "AkF1T_xVT7uDHhSgxUSjOw",
    "version": 137,
    "state_uuid": "Iah-PnbTQ4OAdvK7lS2mjQ",
    "master_node": "NXLq-qizQNe8GX2cLqI2EQ",
    "blocks": {},
    "nodes": {
      "NXLq-qizQNe8GX2cLqI2EQ": {
        "name": "NXLq-qi",
        "ephemeral_id": "z-diXsm9SOO48wuEmPqnxg",
        "transport_address": "172.31.9.100:9300",
        "attributes": {
```

Elasticsearch 클러스터 운영 - Shard Allocation

Cluster Reroute More..
https://www.elastic.co/guide/en/elasticsearch/reference/current/cluster-reroute.html

Elasticsearch 클러스터 운영 - Shard Allocation

PUT _cluster/settings 의 disk threshold 를 이용하는 방법

```
PUT _cluster/settings
{
   "transient":
   {
      "cluster.routing.allocation.disk.threshold_enabled": "true",
      "cluster.routing.allocation.disk.watermark.low": "85%",
      "cluster.routing.allocation.disk.watermark.high": "90%",
      "cluster.routing.allocation.disk.watermark.flood_stage": "95%"
   }
}
```


```
{
  "acknowledged" : true,
  "persistent" : { },
  "transient" : {
    "cluster" : {
      "routing" : {
        "allocation" : {
          "disk" : {
            "threshold_enabled" : "true",
            "watermark" : {
              "low" : "85%",
              "flood_stage" : "95%",
              "high" : "90%"
            }
          }
        }
      }
    }
  }
}
```


watermark.low - 더 이상 차오르지 못하도록 할 임계치
신규로 생성되는 인덱스는 제외

watermark.high - 설정 즉시 임계치 이상 되는 노드를 임계치로
맞추기 위해 샤드 재분배 진행

Elasticsearch 클러스터 운영 - Shard Allocation

watermark.flood_stage
- 디스크 용량이 더 이상 차오르지 못하도록 할 임계치
- 임계치가 넘으면 인덱스를 삭제 가능한 read only 모드로 변경
- 데이터 정리 후 해당 인덱스에 대해 read only 해제 필요

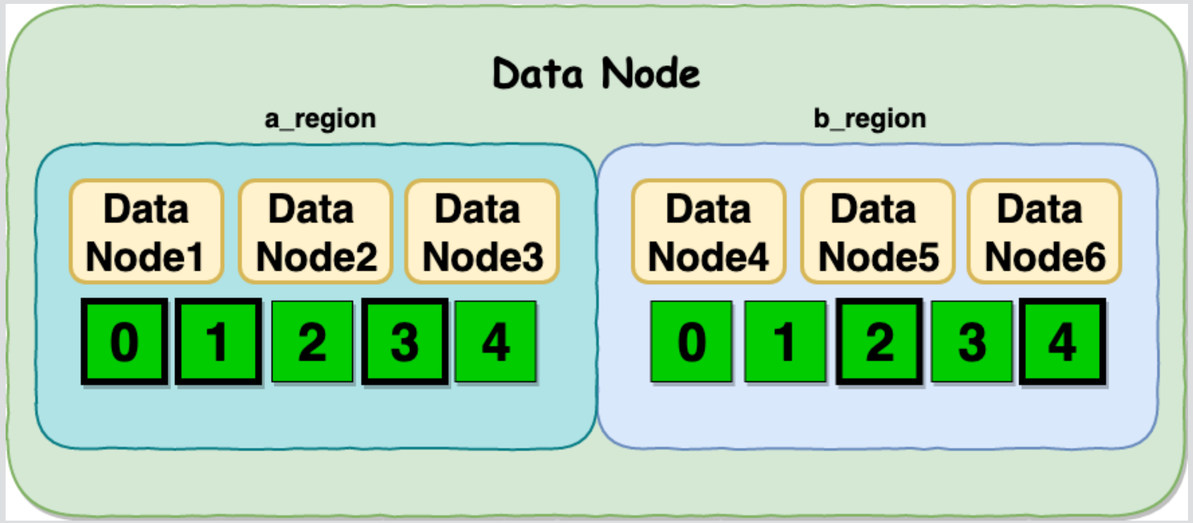
```
PUT twitter/_settings
{
    "index.blocks.read_only_allow_delete": null
}
```

Elasticsearch 클러스터 운영 - Shard Allocation

데이터 노드 그룹으로 샤드 할당하기

cluster.routing.allocation.awareness.attributes
- 마스터 노드에 세팅하는 순간부터 적용

node.attr.rack_id: a_region
node.attr.rack_id: b_region
- 데이터 노드의 region 군을 세팅



마스터노드가 awareness 하는 순간부터 샤드를 데이터 노드에 세팅된 rack_id 를 기준으로 rebalancing 시작

하나의 리전에서 서비스하다가 이중화가 필요할 때에 데이터노드를 먼저 세팅 후 마스터 노드에서 세팅

Elasticsearch 클러스터 운영 - Shard Allocation

Disk-based Shard Allocation More..
https://www.elastic.co/guide/en/elasticsearch/reference/current/disk-allocator.html

Elasticsearch 클러스터 운영 - Index setting

인덱스의 settings

1) Static index settings
- number_of_shards

2) Dynamic index settings
- number_of_replicas
- refresh_interval
- index.routing.allocation.enable

3) other settings..
- Analysis, Mapping, Slowlog..

Elasticsearch 클러스터 운영 - Index setting

Dynamic index settings
- 운영중에 인덱스 세팅을 변경
- RestAPI 로 변경사항을 요청

- number_of_replicas - 운영중에 리플리카 샤드 개수를 변경

```
PUT twitter/_settings
{
    "index.number_of_replicas" : 2
}
```

Elasticsearch 클러스터 운영 - Index setting

refresh_interval
- 인덱싱이 될 때 데이터가 메모리 버퍼캐시로 쓰인 이후 검색이 가능하도록 디스크로 쓰이는 시간 텀
- reset 은 null 로 설정

```
PUT twitter/_settings
{
    "index.refresh_interval" : "2s"
}

GET twitter/_settings
```


```
{
  "acknowledged": true
}
```



```
{
  "twitter": {
    "settings": {
      "index": {
        "refresh_interval": "2s",
        "number_of_shards": "3",
        "provided_name": "twitter",
        "creation_date": "1539620123793",
        "number_of_replicas": "2",
        "uuid": "woejXzXgTT-3TMGV3GBe5A",
        "version": {
          "created": "6040299"
        }
      }
    }
  }
}
```

Elasticsearch 클러스터 운영 - Index setting

운영중인 모든 인덱스에 세팅 일괄 적용하기
- 인덱스 이름 대신 _all 로 세팅

```
PUT _all/_settings
{
   "index.refresh_interval" : "2s"
}

GET twitter/_settings
```


```
{
  "acknowledged": true
}
```



```
{
  "twitter": {
    "settings": {
      "index": {
        "refresh_interval": "2s",
        "number_of_shards": "3",
        "provided_name": "twitter",
        "creation_date": "1539620123793",
        "number_of_replicas": "2",
        "uuid": "woejXzXgTT-3TMGV3GBe5A",
        "version": {
          "created": "6040299"
        }
      }
    }
  }
}
```

Elasticsearch 클러스터 운영 - Index setting

Routing Allocation

- 새롭게 할당된 데이터 노드에 대해 샤드를 재할당하는 방식 결정

- all (default) - 모든 샤드들에게 할당을 허용
- none - 샤드가 할당되지 않도록 설정
- primaries - 프라이머리 샤드만 할당되도록 설정
- new_primaries - 새롭게 생성되는 인덱스의 프라이머리 샤드만 할당되도록 설정
- null - default 로 설정

```
PUT twitter/_settings
{
    "index.routing.allocation.enable" : null
}
```

```
{
  "acknowledged": true
}
```

```
{
  "twitter": {
    "settings": {
      "index": {
        "routing": {
          "allocation": {
            "enable": "none"
          }
        },
        "number_of_shards": "5",
        "provided_name": "twitter",
        "creation_date": "1539751544923",
        "number_of_replicas": "1",
        "uuid": "yAKS9hdvR86AzWF3w48U-g",
        "version": {
          "created": "6040199"
        }
      }
    }
  }
}
```

Elasticsearch 클러스터 운영 - Index setting

Routing Rebalance
- 데이터 노드에 샤드를 어떤 방식으로 재배치할 것인지를 결정
- assigned 샤드를 대상으로 재할당 방식 결정

- all (default) - 모든 샤드들에게 재배치 허용
- none - 샤드가 재배치되지 않도록 설정
- primaries - 프라이머리 샤드만 재배치되도록 설정
- replicas - 리플리카 샤드만 재배치되도록 설정
- null - default 로 설정

```
PUT twitter/_settings
{
   "index.routing.rebalance.enable" : null
}
```

```
{
  "acknowledged": true
}
```

```
{
  "twitter": {
    "settings": {
      "index": {
        "routing": {
          "rebalance": {
            "enable": "none"
          },
          "allocation": {
            "enable": "none"
          }
        },
        "refresh_interval": "2s",
        "number_of_shards": "3",
        "provided_name": "twitter",
        "creation_date": "1539620123793",
        "number_of_replicas": "2",
        "uuid": "woejXzXgTT-3TMGV3GBe5A",
        "version": {
          "created": "6040299"
        }
      }
    }
  }
}
```

Elasticsearch 클러스터 운영 - Index setting

Mapping
- 문서가 인덱싱 될 때 문서와 문서에 포함된 필드들을 어떻게 저장할지를 결정하는 과정
- 6.x 버전부터 Multi Mapping Deprecated

1) Dynamic Mapping
- Elasticsearch 가 인입되는 도큐멘트를 보고 알아서 타입을 찾아 매핑

2) Static Mapping
- 사용자가 정의한 스키마를 기준으로 매핑

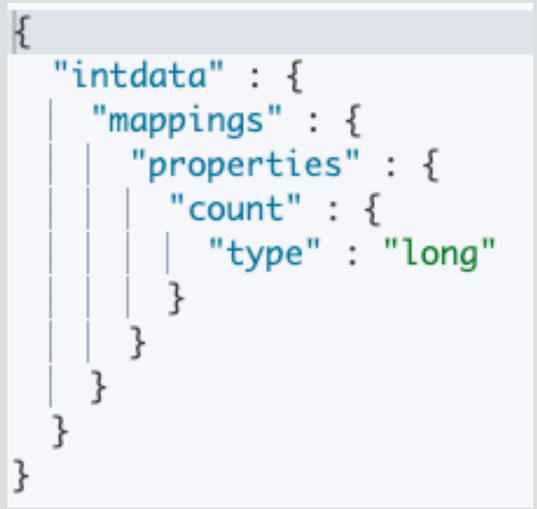
Elasticsearch 클러스터 운영 - Index setting

```
PUT intdata/_doc/1
{ "count": 5 }

GET intdata/_mapping

PUT strdata/_doc/1
{ "stringdata": "strdata" }

GET strdata/_mapping
```



Dynamic Mapping 으로 인덱스에 mappings 가 정의되고,
count key 는 long 타입으로 자동 매핑되어 매핑 생성

Elasticsearch 클러스터 운영 - Index setting

Dynamic Mapping Field

Value Type	Mapping Field / Description
null	No field is added
true of false	boolean field
floating point number	float field
integer	long field
object	object field
array	string or object field
date string	double or long field
text string	text field with a keyword sub-field

Elasticsearch 클러스터 운영 - Index setting

Index Settings More..
https://www.elastic.co/guide/en/elasticsearch/reference/current/index-modules.html#index-modules-settings

Elasticsearch 클러스터 운영 - Template

Template
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
- order 가 높은 번호가 낮은 번호를 override 하여 merging

```
PUT _template/mytemplate
{
   "index_patterns": ["te*", "bar*"],
   "order" : 0,
   "settings": {
      "index.number_of_shards": 1
   }
}

GET _template/mytemplate
```

```
{
  "mytemplate": {
    "order": 1,
    "index_patterns": [
      "te*",
      "bar*"
    ],
    "settings": {
      "index": {
        "number_of_shards": "1"
      }
    },
    "mappings": {},
    "aliases": {}
  }
}
```

Elasticsearch 클러스터 운영 - Template

- 템플릿 삭제는 DELETE Method 이용

```
DELET _template/mytemplate
```

Template More..
https://www.elastic.co/guide/en/elasticsearch/reference/current/indices-templates.html

Elasticsearch 클러스터 운영 - Hot-data / Warm-data

Hot-data / Warm-data

- ES 에서는 빠른 응답을 위해 SSD 디스크를 사용하는 것을 권고
- 인덱스가 크고 보관기간이 길면 비용 부담이 증가
- 최근 데이터를 더 자주 보는 경향을 이용한 매커니즘
- 상대적으로 비용이 저렴하고 고용량인 SATA 디스크를 이용
- elasticsearch.yml 파일, Template, Curator 를 이용하여 운영
- 최근 데이터 주기를 정하여 시간이 지난 인덱스의 샤드를 warm data node 쪽으로 재할당하는 방식

Elasticsearch 클러스터 운영 - Hot-data / Warm-data

- 각각의 노드가 어떤 타입으로 운영되는지를 설정하여 시작

$ sudo vi /etc/elasticsearch/elasticsearch.yml

```
# hotdata node setting
node.attr.box_type: hot

# warmdata node setting
node.attr.box_type: warm
```

Elasticsearch 클러스터 운영 - Hot-data / Warm-data

- 템플릿을 통해 패턴에 적용되는, 새로 생성되는 인덱스의 샤드를 hot 쪽으로만 할당

```
PUT _template/mytemplate
{
   "index_patterns": ["*"],
   "order" : 0,
   "settings": {
      "index.number_of_shards": 1,
      "index.routing.allocation.require.box_type" : "hot"
   }
}
```

- 이후, hot node 에 보관하는 일자 이후의 인덱스들의 샤드들을 warm node 로 재할당

- 위 과정은 사람이 일일히 하기 힘들어 curator 를 두어 배치로 운영

Elasticsearch 클러스터 운영 - Hot-data / Warm-data

- index.routing.allocation.require.box_type 을 이용해 Hot / Warm 간 샤드 이동

```
PUT test/_settings
{
      "index.routing.allocation.require.box_type" : "warm"
}
```

Elasticsearch 클러스터 운영 - Hot-data / Warm-data

Hot/Warm Data More..
https://www.elastic.co/blog/hot-warm-architecture-in-elasticsearch-5-x

실습

Elasticsearch Hot Warm Data Node
https://github.com/benjamin-btn/ES7-Tutorial/tree/master/ES-Tutorial-4

Elasticsearch API 활용하기

Cluster API - 클러스터 운영 API 다뤄보기

Reindex API - 데이터 마이그레이션

Bulk API - 도큐멘트 한번에 인덱싱하기

그 외 운영에 유용한 API

Elasticsearch API 활용하기

Elasticsearch 는 여러가지 API 를 두어 온라인 상의 사용이나 운영의 편이를 도모

- 운영을 위해 클러스터 상태나 지표들을 볼 수 있는 모니터링 API
- 클러스터의 설정을 변경할 수 있는 클러스터 설정변경 API
- 데이터를 이관하거나 별칭을 달 수 있는 API 등

Elasticsearch API 활용하기 - Cluster API

Cluster API

- 운영중인 클러스터의 세팅정보 확인이나 온라인 상태로 설정을 변경할 수 있는 API
- 자주 변경할 여지가 있는 사항들은 cluster api 로 진행
- 설정 모드는 두 가지로 나뉨

Elasticsearch API 활용하기 - Cluster API

1) Transient
- Full cluster restart 시 리셋되는 설정

2) Persistent
- 사용자가 변경하지 않으면 영구적으로 보존되는 설정
- static setting 보다 우선순위가 높음

Elasticsearch API 활용하기 - Cluster API

현재 클러스터 세팅 확인하기

```
GET _cluster/settings
```

클러스터 세팅 초기화

```
PUT _cluster/settings
{
   "persistent" : {
     "cluster.routing.allocation.disk.threshold_enabled": null
   },
   "transient" : {
       "cluster.routing.allocation.enable" : null
   }
}
```


```
{
  "acknowledged" : true,
  "persistent" : {
    "cluster" : {
      "routing" : {
        "allocation" : {
          "disk" : {
            "threshold_enabled" : "true"
          }
        }
      }
    }
  },
  "transient" : {
    "cluster" : {
      "routing" : {
        "allocation" : {
          "enable" : "new_primaries"
        }
      }
    }
  }
}
```

Elasticsearch API 활용하기 - Cluster API

_cluster API 로 운영중인 특정 노드의 샤드 제외
- 좀 더 안정적인 롤링 리스타트를 할 때 샤드를 미리 다 제거하고 진행
- unassigned 샤드가 있는 상황에서 추가로 노드를 작업해야할 때 진행
- 아이피는 class 별로도 세팅 가능 (ex. 1.1.1.0/24)

```
PUT _cluster/settings
{
   "transient" : {
      "cluster.routing.allocation.exclude._ip" : "1.1.1.1, 2.2.2.2, 3.3.3.*"
   }
}
```

_name - node name 기준으로도 exclude 가능
_host - host name 기준으로도 exclude 가능

Elasticsearch API 활용하기 - Cluster API

POST _cluster/reroute 를 이용한 샤드 할당에 실패한 샤드 강제 분배

- 인덱스 세팅 중 어떤 상황에 의해 할당되지 못한 샤드를 다시 할당하는 시도 횟수에 제한이 있음
- index.allocation.max_retries 값에 의해 default 로 5번만 추가 시도
- 5번 전부 실패하면 샤드 할당을 더이상 하지 않음
- 대표적인 예로 디스크 볼륨이 부족한 경우에 5번 시도 후 샤드 할당을 포기
- 디스크 볼륨을 정리하고 retry 를 시도

```
POST _cluster/reroute?retry_failed
```

POST _cluster/allocation/explain 을 통해 샤드가 왜 할당되지 못했는지를 확인

- 용량 문제가 아닌데 retry 가 시도된 후 복구가 되지 않을 때에는 해당 명령으로 원인을 확인

```
POST _cluster/allocation/explain
```

Elasticsearch API 활용하기 - Cluster API

모든 인덱스에 대해 _all 이나 wildcard 를 대상으로 삭제작업 방지하기

- 인덱스가 정의될 위치에 _all 을 넣거나 wildcard (*) 를 넣으면 전체 인덱스에 대해 작업 가능
- DELETE 의 경우는 의도되지 않은 실수를 방지하기 위해 해당 작업 disable 가능

```
PUT _cluster/settings
{
  "transient": {
      "action.destructive_requires_name": true
  }
}
```

Elasticsearch API 활용하기 - Cluster API

Cluster API More..

https://www.elastic.co/guide/en/elasticsearch/reference/current/cluster-update-settings.html
https://www.elastic.co/guide/en/elasticsearch/reference/current/shards-allocation.html
https://www.elastic.co/guide/en/elasticsearch/reference/current/allocation-filtering.html
https://www.elastic.co/guide/en/elasticsearch/reference/current/cluster-reroute.html
https://www.elastic.co/guide/en/elasticsearch/reference/current/cluster-allocation-explain.html
https://www.elastic.co/guide/en/elasticsearch/reference/current/indices-delete-index.html

Elasticsearch API 활용하기 - Reindex API

POST _reindex 를 이용한 재색인
- 인덱스를 복제할 때 사용
- 원본 인덱스의 세팅이나 매핑은 복제되지 않음
- 클러스터 내부 뿐 아니라 외부 클러스터의 인덱스도 복제 가능

```
POST _reindex
{
    "source": {
        "index": "twitter"
    },
    "dest": {
        "index": "new_twitter"
    }
}
```

Elasticsearch API 활용하기 - Reindex API

외부 클러스터에서 reindex 가능
인덱스를 복제 받은 클러스터 elasticsearch.yml 에 원본 인덱스 클러스터를 whitelist 로 등록

```
reindex.remote.whitelist: "1.1.1.1:9200"

$ curl -XPOST -H 'Content-Type: application/json' http://{my_cluster_url}/_reindex
{
    "source": {
        "remote": {
            "host": "http://1.1.1.1:9200"
        },
        "index": "twitter"
    },
    "dest": {
        "index": "re_twitter"
    }
}
```

Elasticsearch API 활용하기 - Reindex API

Reindex API More..
https://www.elastic.co/guide/en/elasticsearch/reference/current/docs-reindex.html

Elasticsearch API 활용하기 - Bulk API

- 인덱스 문서의 인덱싱, 삭제, 업데이트를 벌크로 진행할 수 있는 API
- Java, Python, Perl 등 언어별로 bulk api 라이브러리 제공

```
POST _bulk
{ "index" : { "_index" : "test", "_type" : "_doc", "_id" : "1" } }
{ "field1" : "value1" }
{ "delete" : { "_index" : "test", "_type" : "_doc", "_id" : "2" } }
{ "create" : { "_index" : "test", "_type" : "_doc", "_id" : "3" } }
{ "field1" : "value3" }
{ "update" : {"_id" : "1", "_type" : "_doc", "_index" : "test"} }
{ "doc" : {"field2" : "value2"} }
```

Elasticsearch API 활용하기 - Bulk API

json file 형태의 문서도 bulk api 를 활용하여 처리 가능

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

Elasticsearch API 활용하기 - Bulk API

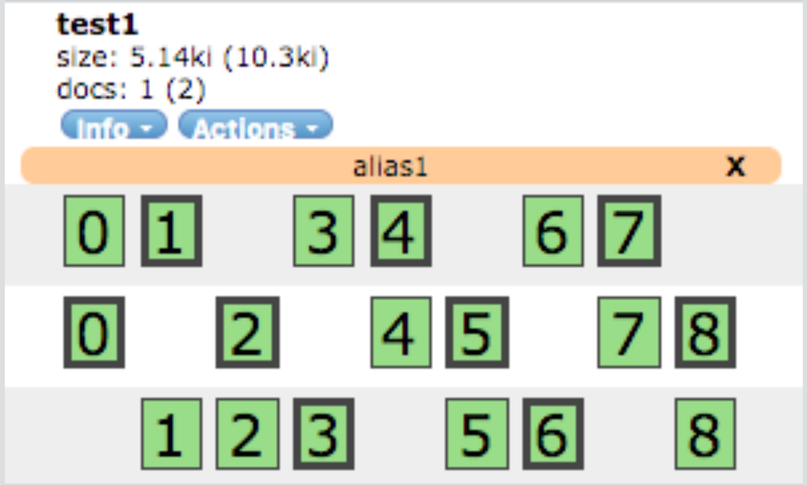
Bulk API More..
https://www.elastic.co/guide/en/elasticsearch/reference/current/docs-bulk.html

Elasticsearch API 활용하기 - 그 외 운영에 유용한 API

_aliases API
- 인덱스에 별칭을 부여하는 API
- _reindex API 와 함께 자주 사용
- 존재하는 인덱스와 같은 이름으로는 설정 불가능

```
POST /_aliases
{
    "actions": [
        { "add": { "index": "test1", "alias": "alias1" } }
    ]
}
```

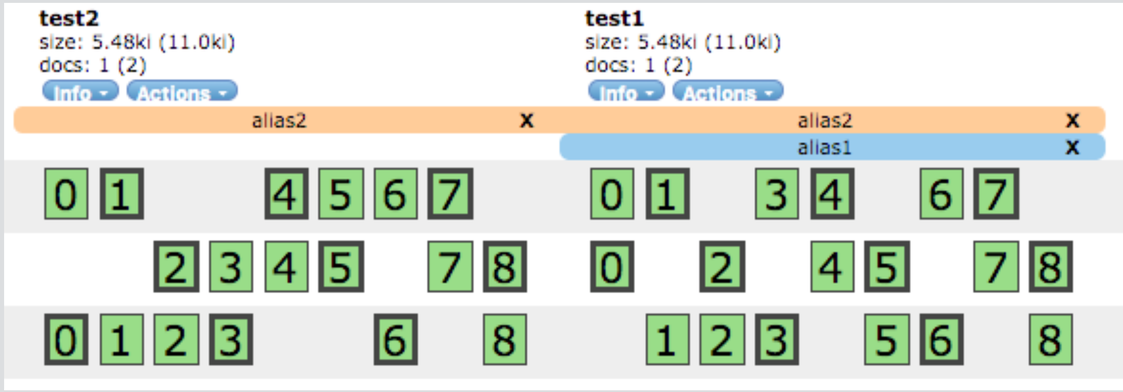
```
POST /_aliases
{
    "actions": [
        { "remove": { "index": "test1", "alias": "alias1" } }
    ]
}
```



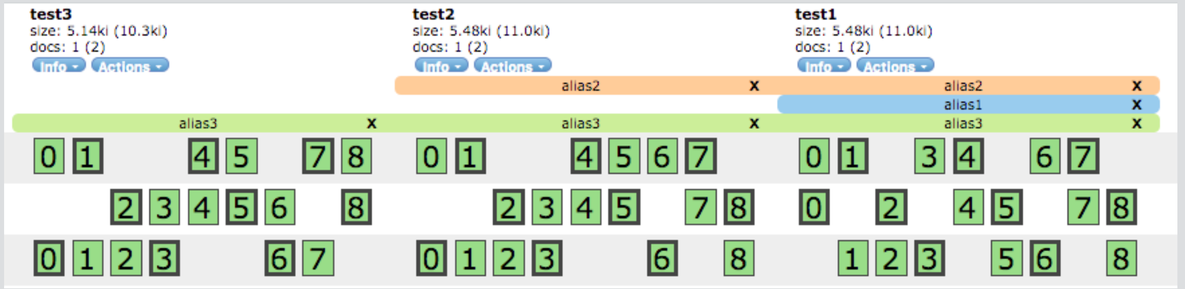
Elasticsearch API 활용하기 - 그 외 운영에 유용한 API

- 여러개의 인덱스에 걸쳐 설정 가능

```
POST /_aliases
{
    "actions": [
        { "add": { "indices": ["test1", "test2"], "alias":
"alias2" } }
    ]
}
```




```
POST /_aliases
{
    "actions": [
        { "add": { "index": "test*", "alias": "alias3" } }
    ]
}
```



Elasticsearch API 활용하기 - 그 외 운영에 유용한 API

_forcemerge API

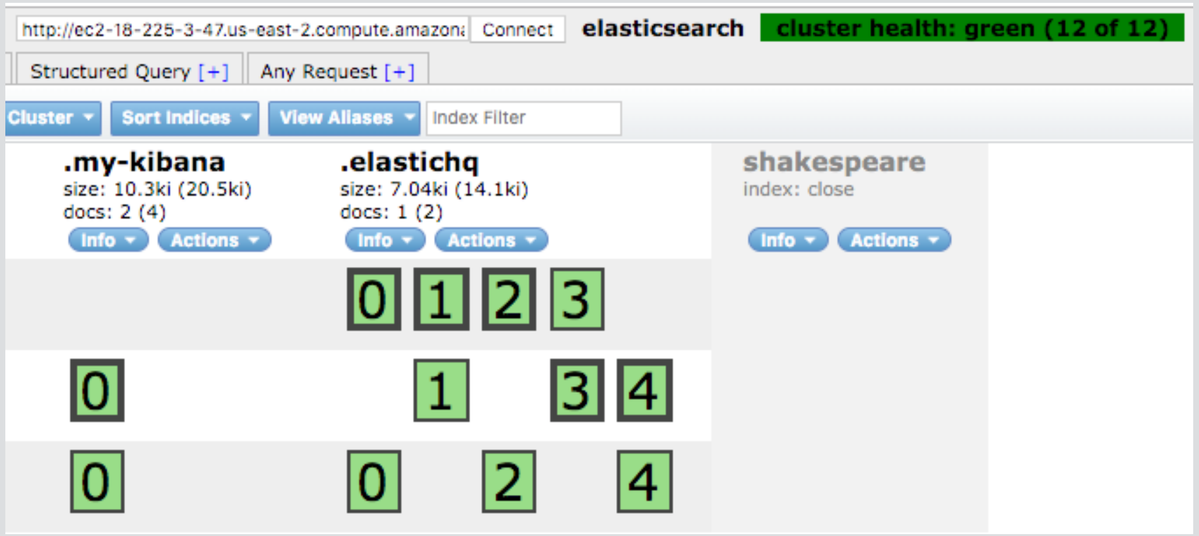
- segment 를 강제로 병합하는 API
- 인덱싱 중인 인덱스에는 사용 비추
- 인덱싱이 끝난 인덱스는 하나의 segment 로 merge 를 추천
- I/O 비용이 크기 때문에 인덱싱이나 검색이 없는 시간대에 진행

```
POST /_forcemerge?max_num_segments=1
```

Elasticsearch API 활용하기 - 그 외 운영에 유용한 API

_open/close API
- 인덱스의 상태를 open/close 할 수 있는 API
- closed 된 인덱스는 read/write 불가
- 클러스터 전체 샤드에서 제외
- 라우팅 disabled

```
POST twitter/_close
POST twitter/_open
```



Elasticsearch API 활용하기 - 그 외 운영에 유용한 API

ETC APIs..
https://www.elastic.co/guide/en/elasticsearch/reference/current/indices-aliases.html
https://www.elastic.co/guide/en/elasticsearch/reference/current/indices-forcemerge.html
https://www.elastic.co/guide/en/elasticsearch/reference/current/indices-open-close.html

Q & A