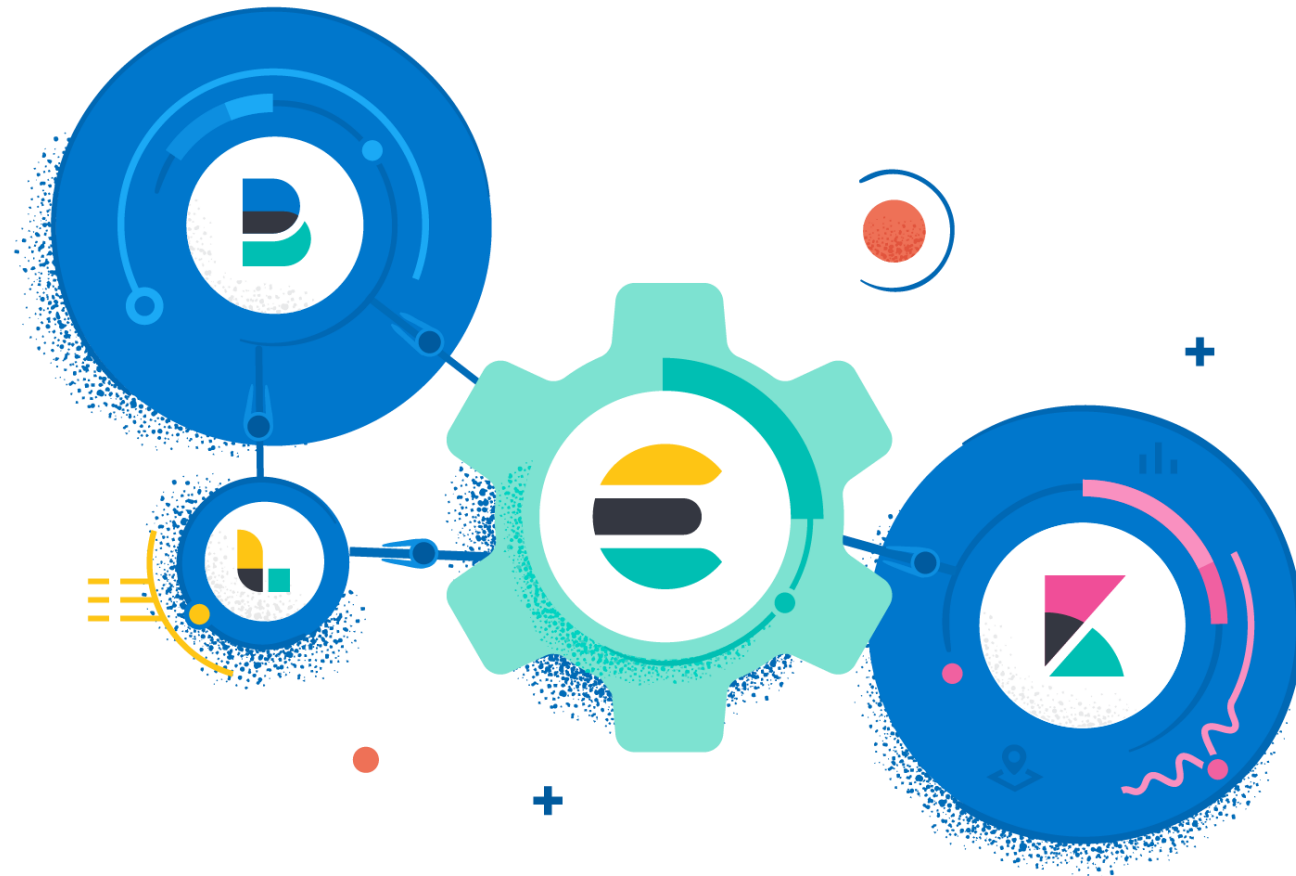


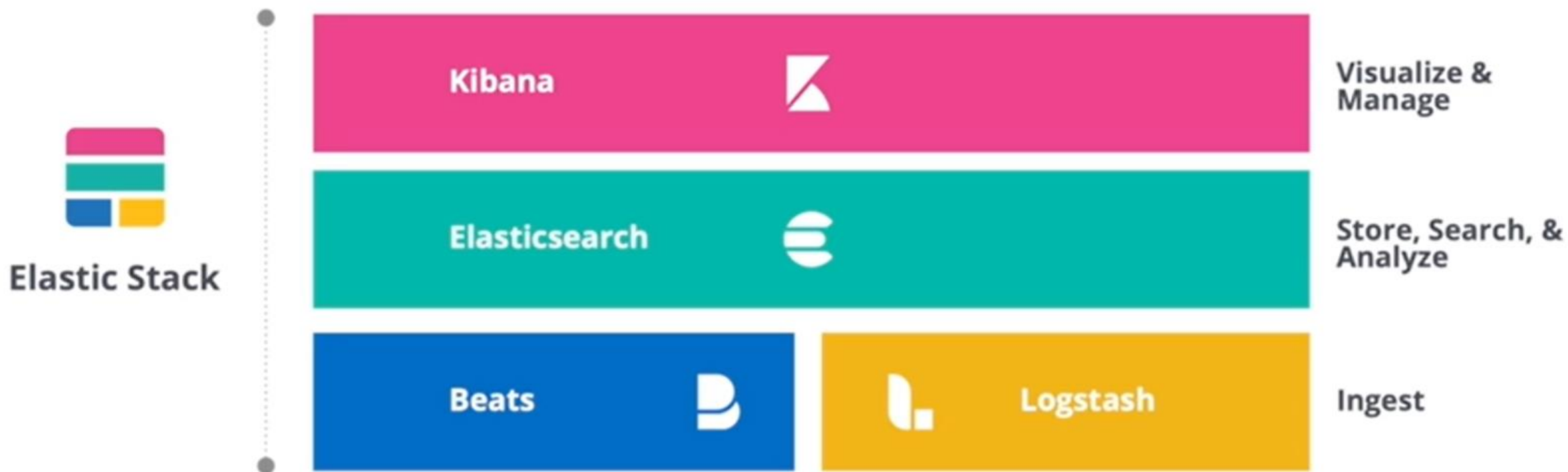
# 빅데이터 검색엔진 Elastic Stack



# Elastic Stack

Elasticsearch, Logstash, Kibana 세가지 오픈 소스 프로젝트가 ELK Stack이라는 명칭으로 서비스가 제공되었고 여기에 Beats를 도입하여 Elastic Stack이라고 합니다.

<https://github.com/elastic>



# Elasticsearch

아파치 루씬 기반의 Full Text 검색이 가능한 오픈소스 분석엔진입니다.  
주로 REST API를 이용해 처리하며, 대량의 데이터를 거의 실시간으로 저장, 검색 및 분석 할 수 있습니다.

Search results for "서울, 한국" (Seoul, Korea) showing hotels and their details.

**검색 결과 요약하기:**  
3,033개 호텔, 주택 및 아파트

**숙박 시설 이름으로 검색**

**숙박 시설 이름**

**인기 필터**

- ☐ 수영장
- ☐ 아침 식사 포함
- ☐ 호텔
- ☐ 주차 포함
- ☐ 객실 내 욕조

**1박 요금:**  
₩0 ~ ₩1,000,000+

**숙박 시설 등급**

**고객 평점: 0 ~ 10**

**서울, 한국**

정렬 기준: 추천, 숙박 시설 등급, 거리, 고객 평점, 가격

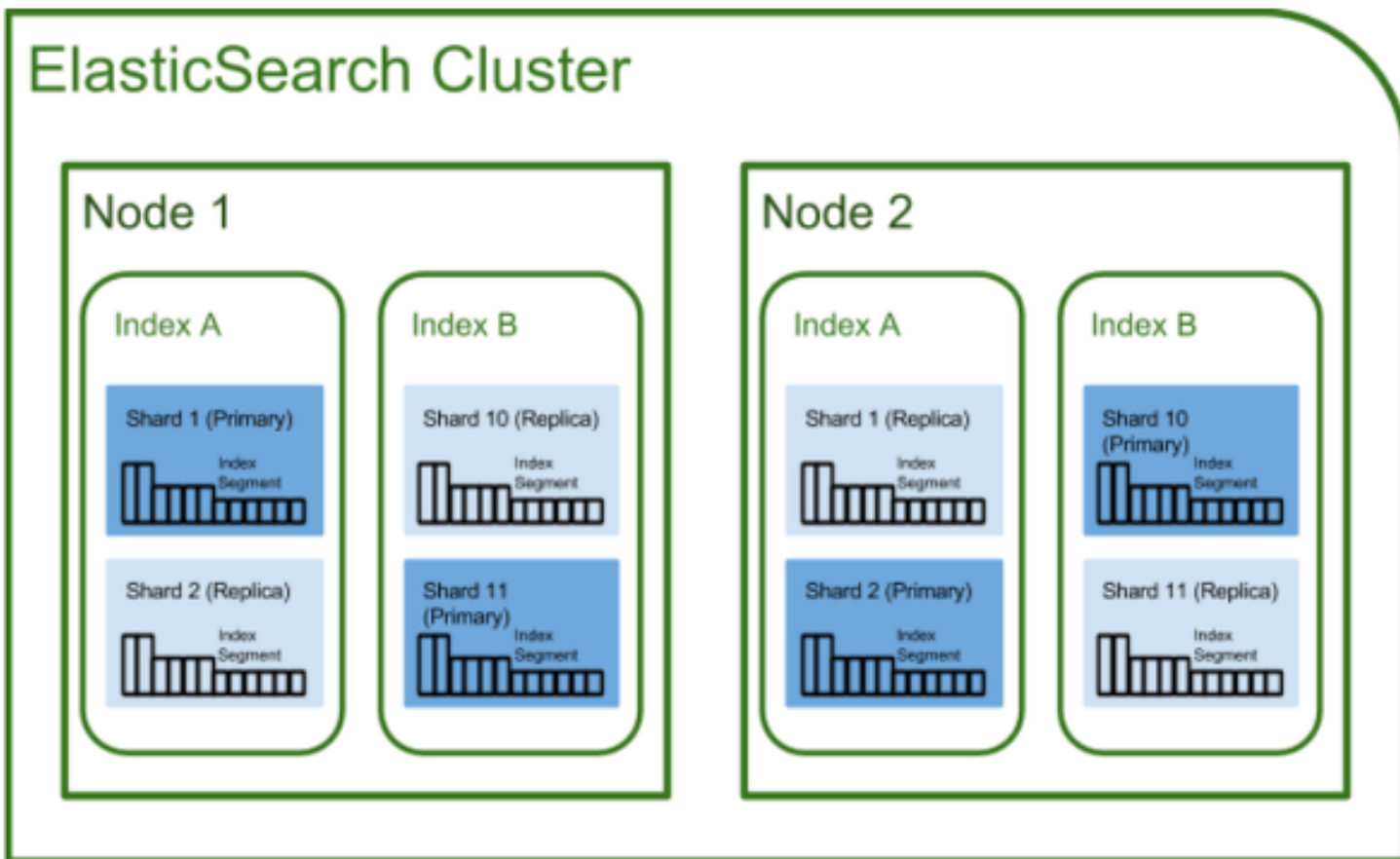
**광고 콘래드 서울 (Conrad Seoul) 5성급**  
₩210,000  
영등포구 국제금융로 10, 서울특별시, 07326  
여의동  
• 서울 (ICN-인천국제공항)까지 43km  
• 서울역까지 5.5km  
9.0 매우 훌륭함  
831개 Hotels.com 고객 이용 후기  
고객's PICK, 수영장, 무료 주차, 반려동물 동반 가능, 스파, 피트니스, 레스토랑  
도착 단 3일 전, 10% 할인가로 예약하고 편안한 여행을 준비하세요.  
다음 호텔 투숙을 10% 할인가로 지금 예약하세요. 계획이 바뀔 수 있다는 점을 이해하기 때문에 도착 72시간 전까지 변경 및 취소가 가능합니다. 예약 시 선불 결제 요함.

**광고 롯데 호텔 월드 (Lotte Hotel World) 5성급**  
₩220,000  
송파구 올림픽로 240, 서울특별시, 138-220, 서울특별시  
송파  
• 서울 (ICN-인천국제공항)까지 58km  
• 서울역까지 12km  
8.8 훌륭함  
883개 Hotels.com 고객 이용 후기  
무료 주차, 레스토랑, 바, 욕조, 인터넷  
롯데호텔 월드 리뉴얼 오픈 (2021. 6. 1).  
새롭게 단장한 롯데호텔 월드의 디럭스, 클럽, 스위트 객실을 만나보세요. 한층 더 고급스러워진 클럽 라운지에서 조식 뷔페, 애프터눈 티 서비스, 신선한 재료를 즉석으로 요리해드리는 해피 아워까지 가득 채워진 서비스를 즐겨보세요.

**서울 신라호텔 (The Shilla Seoul) 5성급**  
₩260,000 ~ ₩234,000 (10% 할인)  
중구 동호로 249, 서울특별시, 04605, 서울특별시  
장충동  
서울 (ICN-인천국제공항)까지 58km



# Elasticsearch 아키텍처



- 역색인(Inverted Index)을 통한 빠른 검색
- 클러스터 구성을 통한 분산처리 및 고가용성
- Replica를 활용한 데이터 안정성 증대
- Shard 분배를 통한 선형적 확장(scale-out)
- RESTful API 지원
- Schemaless 지원
- 인덱스 기반의 타입 및 색인 방식 설정 지원

# Elasticsearch 아키텍처

## ■ Cluster

- 하나 이상의 Elasticsearch 노드들로 구성된 노드의 집합
- 클러스터는 Elasticsearch 시스템을 구성하는 가장 큰 단위로 독립적으로 운용

## ■ Node

- 노드는 Elasticsearch가 실행중인 인스턴스

## ■ Shard

- 데이터를 분산하여 저장하기 위해 Index의 범위를 나눈 것

## ■ Replica

- 분산 환경에서 데이터의 신뢰성을 높이기 위해 여러 노드에 데이터를 복제하여 저장하는 것

# Elasticsearch 논리 아키텍처

## ■ Document

- 하나의 JSON 오브젝트로 elasticsearch 시스템에서 데이터를 구성하는 최소 단위
- 일반적인 row형 데이터베이스에서 하나의 row에 대응되는 개념

## ■ Field

- Document를 구성하는 하나의 key-value pair에 해당
- RDBMS의 열(column)에 대응되는 개념
- Elasticsearch는 schema-less 구조이므로, 하나의 Field에 다양한 타입의 데이터를 저장할 수 있음

## ■ Mapping

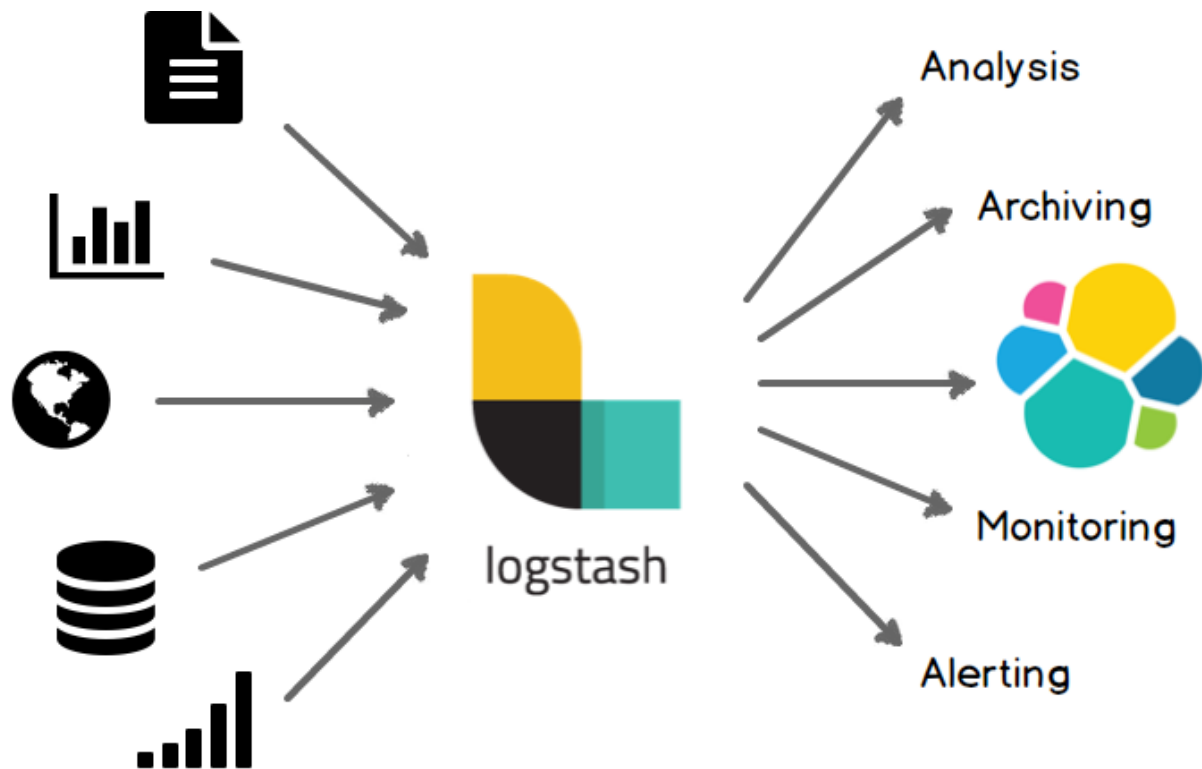
- 하나의 Document를 구성하기 위해 필요한 Field와 Field의 속성, 색인 방법을 정의하는 일련의 과정
- Mapping 과정은 RDBMS에서 스키마를 설계하고 인덱스를 설정하는 과정과 유사

## ■ Indices

- 여러 Type의 집합으로 RDBMS에서 Database에 대응되는 개념
- Index는 Shard와 Replica가 이루어지는 최소 단위

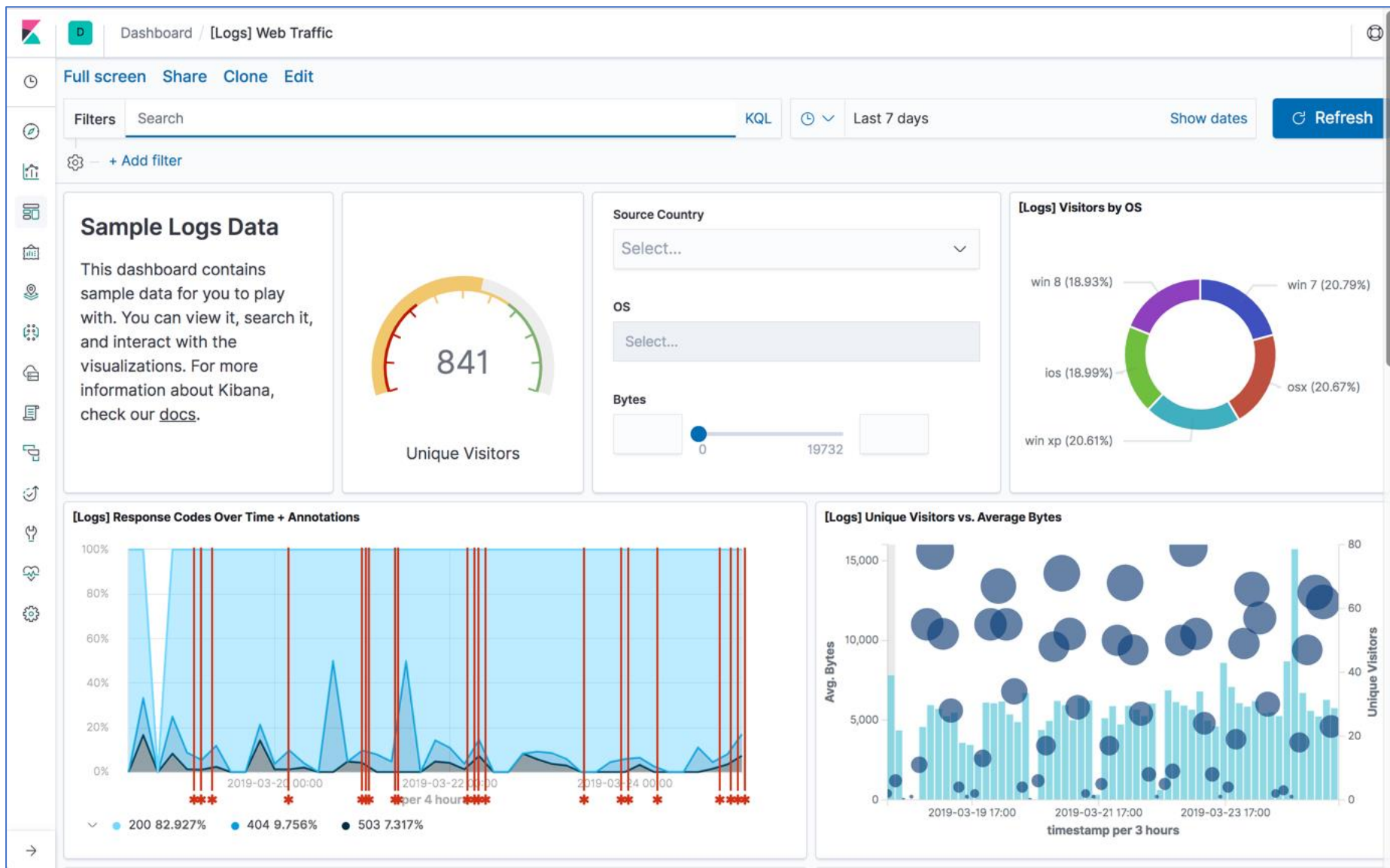
# Logstash

다양한 플러그인을 이용하여 데이터 집계 및 보관, 데이터를 처리하며,  
파이프라인으로 데이터를 수집하여 필터를 통해 변환 후 Elastic Search로 전송합니다.



- **입력** : Beats, Cloudwatch, Eventlog 등의  
다양한 입력을 지원하여 데이터 수집
- **필터** : 형식이나 복잡성에 상관없이 설정을 통해  
데이터를 동적으로 변환
- **출력** : Elastic Search, Email, ECS, Kafka 등  
원하는 저장소에 데이터를 전송

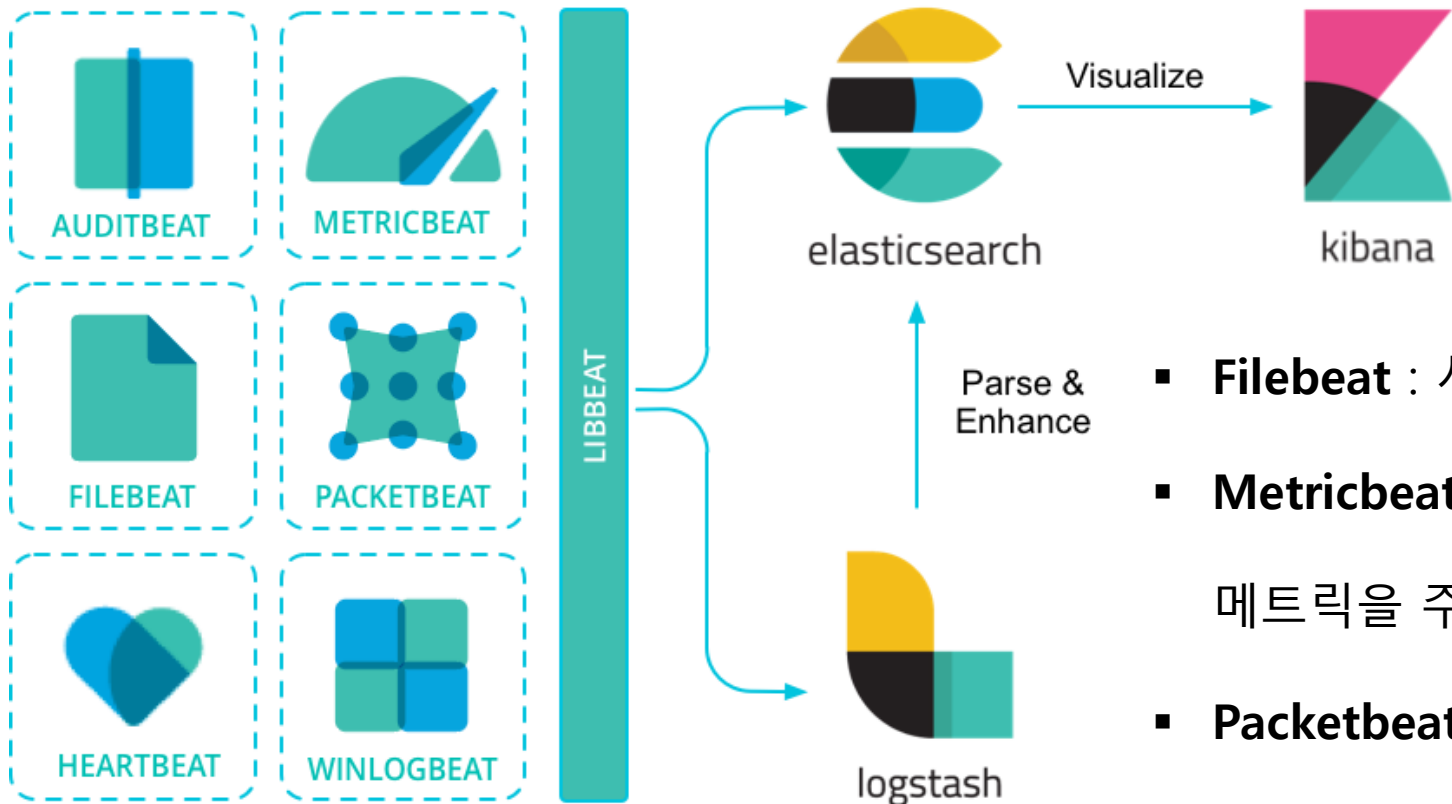
Elasticsearch에서 색인된 데이터를 검색하고 시각화 하는 기능을 제공합니다.





# Beats

경량 에이전트로 설치되어 데이터를 Logstash 또는 Elastic Search로 전송하는 도구입니다.



- **Filebeat** : 서버에서 로그 파일을 제공
- **Metricbeat** : 서버에서 실행중인 운영 체제 및 서비스에서 메트릭을 주기적으로 수집하는 서버 모니터링 에이전트
- **Packetbeat** : 응용 프로그램 서버간에 교환되는 트랜잭션에 대한 정보를 제공하는 네트워크 패킷 분석기
- **Winlogbeat** : Windows 이벤트 로그를 제공

# Elasticsearch 설치

<https://www.elastic.co/kr/downloads/elasticsearch>

Downloads:

↳ [WINDOWS](#) [sha](#) [asc](#)

↳ [MACOS](#) [sha](#) [asc](#)

↳ [LINUX X86\\_64](#) [sha](#) [asc](#)

↳ [LINUX AARCH64](#) [sha](#) [asc](#)

↳ [DEB X86\\_64](#) [sha](#) [asc](#)

↳ [DEB AARCH64](#) [sha](#) [asc](#)

↳ [RPM X86\\_64](#) [sha](#) [asc](#)

↳ [RPM AARCH64](#) [sha](#) [asc](#)

↳ [MSI \(BETA\)](#) [sha](#) [asc](#)

Package Managers:

Install with [yum](#), [dnf](#), or [zypper](#)

Install with [apt-get](#)

Install with [homebrew](#)

---

Containers:

Run with [Docker](#)

---

# Elasticsearch 설치

## ■ jvm.options

- -Xms1g
- -Xmx1g

## ■ elasticsearch.yml

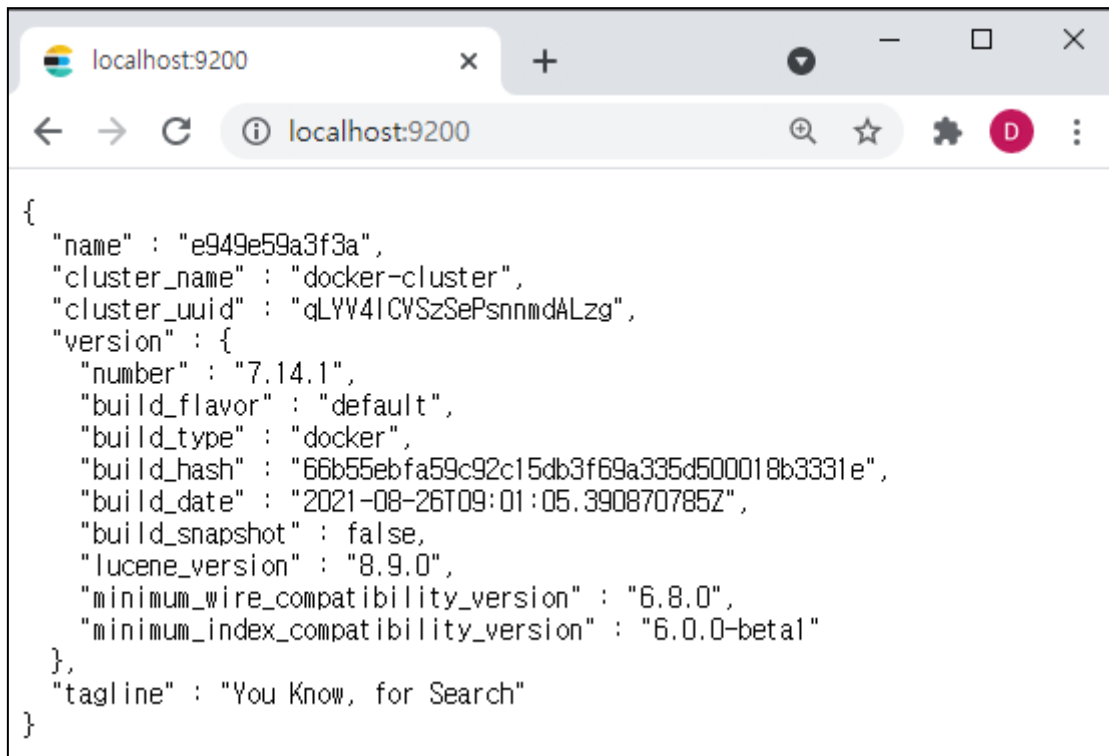
- #network.host: 192.168.0.1
- #http.port: 9200
- #discovery.seed\_hosts: ["host1", "host2"]
- #cluster.initial\_master\_nodes: ["node-1", "node-2"]

## ■ 실행

- cd bin
- elasticsearch

## ■ 실행 확인

- curl localhost:9200
- <http://localhost:9200/> 접속



```
{
  "name" : "e949e59a3f3a",
  "cluster_name" : "docker-cluster",
  "cluster_uuid" : "qLYV4lCVSZePsnmdALzg",
  "version" : {
    "number" : "7.14.1",
    "build_flavor" : "default",
    "build_type" : "docker",
    "build_hash" : "66b55ebfa59c92c15db3f69a335d500018b3331e",
    "build_date" : "2021-08-26T09:01:05.390870785Z",
    "build_snapshot" : false,
    "lucene_version" : "8.9.0",
    "minimum_wire_compatibility_version" : "6.8.0",
    "minimum_index_compatibility_version" : "6.0.0-beta1"
  },
  "tagline" : "You Know, for Search"
}
```

# Kibana 설치

<https://www.elastic.co/kr/downloads/kibana>

Downloads:

↳ [WINDOWS](#) [sha](#) [asc](#)

↳ [MACOS](#) [sha](#) [asc](#)

↳ [LINUX X86\\_64](#) [sha](#) [asc](#)

↳ [LINUX AARCH64](#) [sha](#) [asc](#)

↳ [DEB X86\\_64](#) [sha](#) [asc](#)

↳ [DEB AARCH64](#) [sha](#) [asc](#)

↳ [RPM X86\\_64](#) [sha](#) [asc](#)

↳ [RPM AARCH64](#) [sha](#) [asc](#)

↳ [MSI \(BETA\)](#) [sha](#) [asc](#)

Package Managers:

Install with [yum](#), [dnf](#), or [zypper](#)

Install with [apt-get](#)

Install with [homebrew](#)

Containers:

Run with [Docker](#)

# Kibana 설치

## ■ kibana.yml

- #elasticsearch.hosts: ["http://localhost:9200"]

## ■ 실행

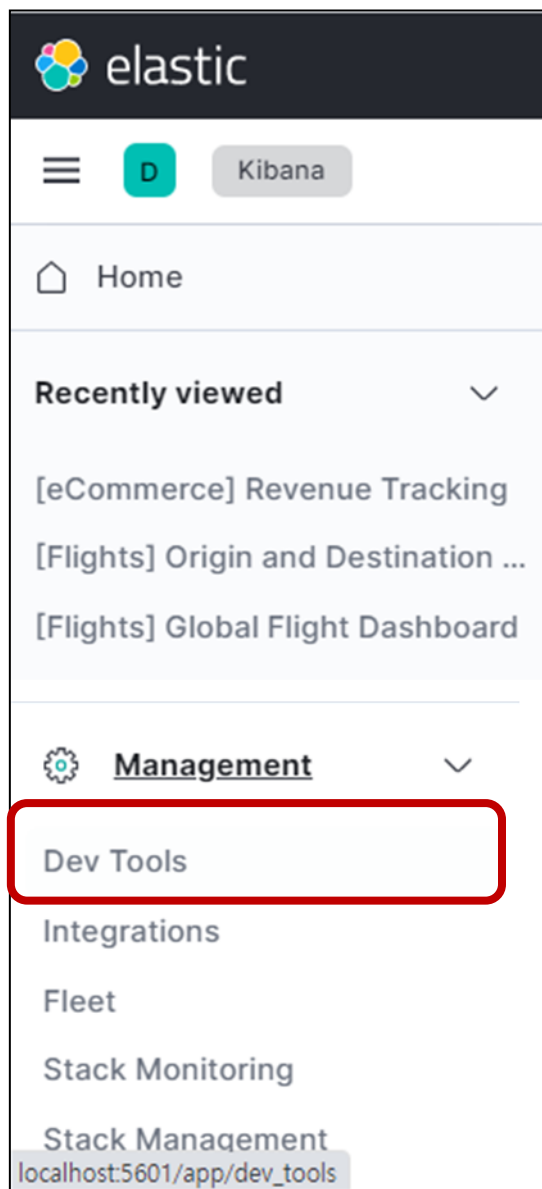
- cd bin
- kibana

## ■ 사용

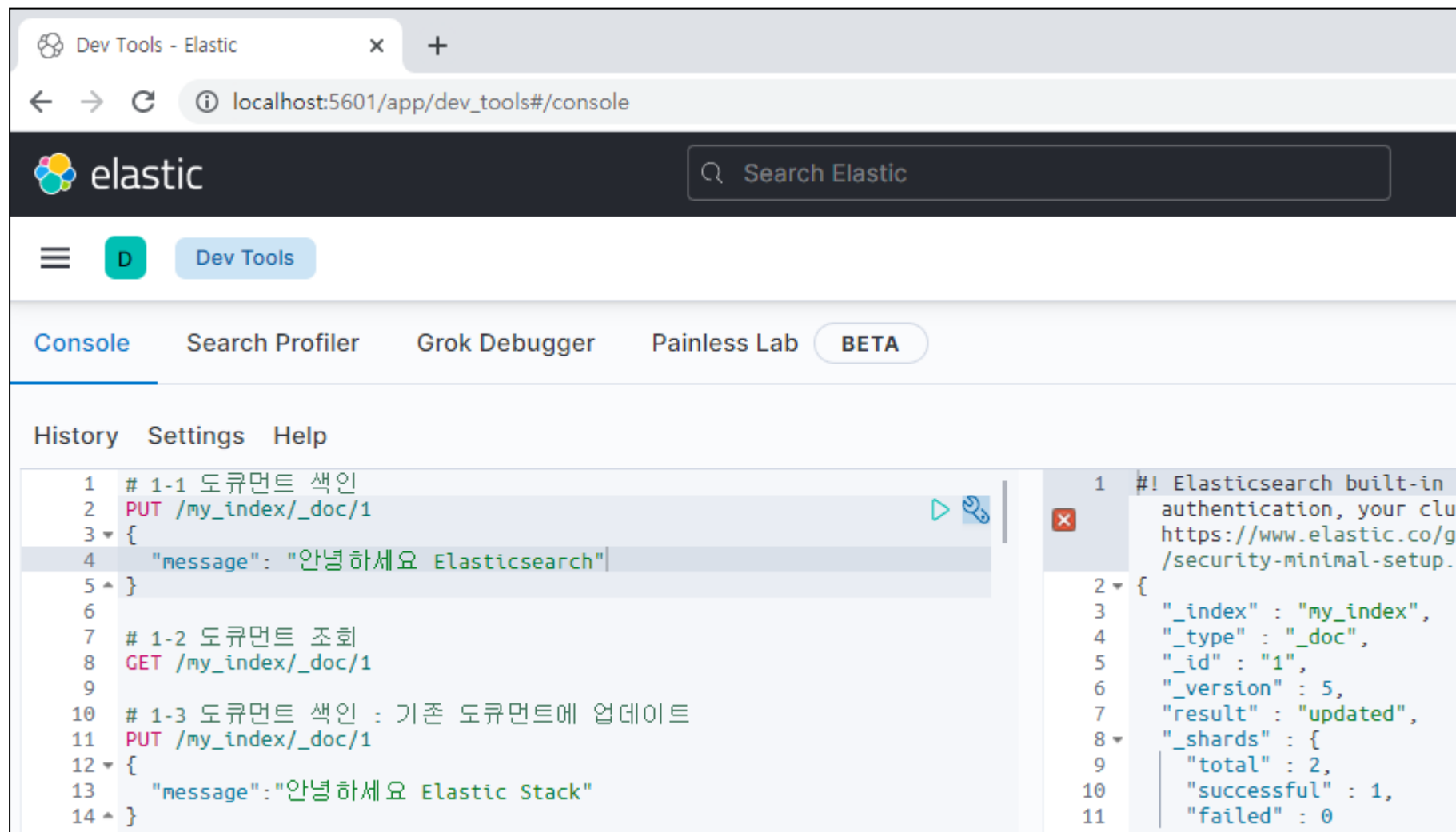
- http://localhost:5601/ 접속

The image displays two screenshots of the Kibana web interface. The left screenshot shows the 'Home' page with three main cards: 'Enterprise Search', 'Observability', and 'Security'. At the bottom, under 'Ingest your data', there is an 'Add data' button highlighted with a red box and labeled '1'. A blue arrow points from this button to the right screenshot. The right screenshot shows the 'Add data' page, which has a top navigation bar with tabs: 'All', 'Logs', 'Metrics', 'Security', 'Sample data', and 'Upload file'. The 'Sample data' tab is highlighted with a red box and labeled '2'. Below the tabs, there are three sample data dashboards: 'Sample eCommerce orders', 'Sample flight data', and 'Sample web logs'. Each dashboard has an 'Add data' button at the bottom, with the first one highlighted by a red box and labeled '3'.

# Elasticsearch 실습



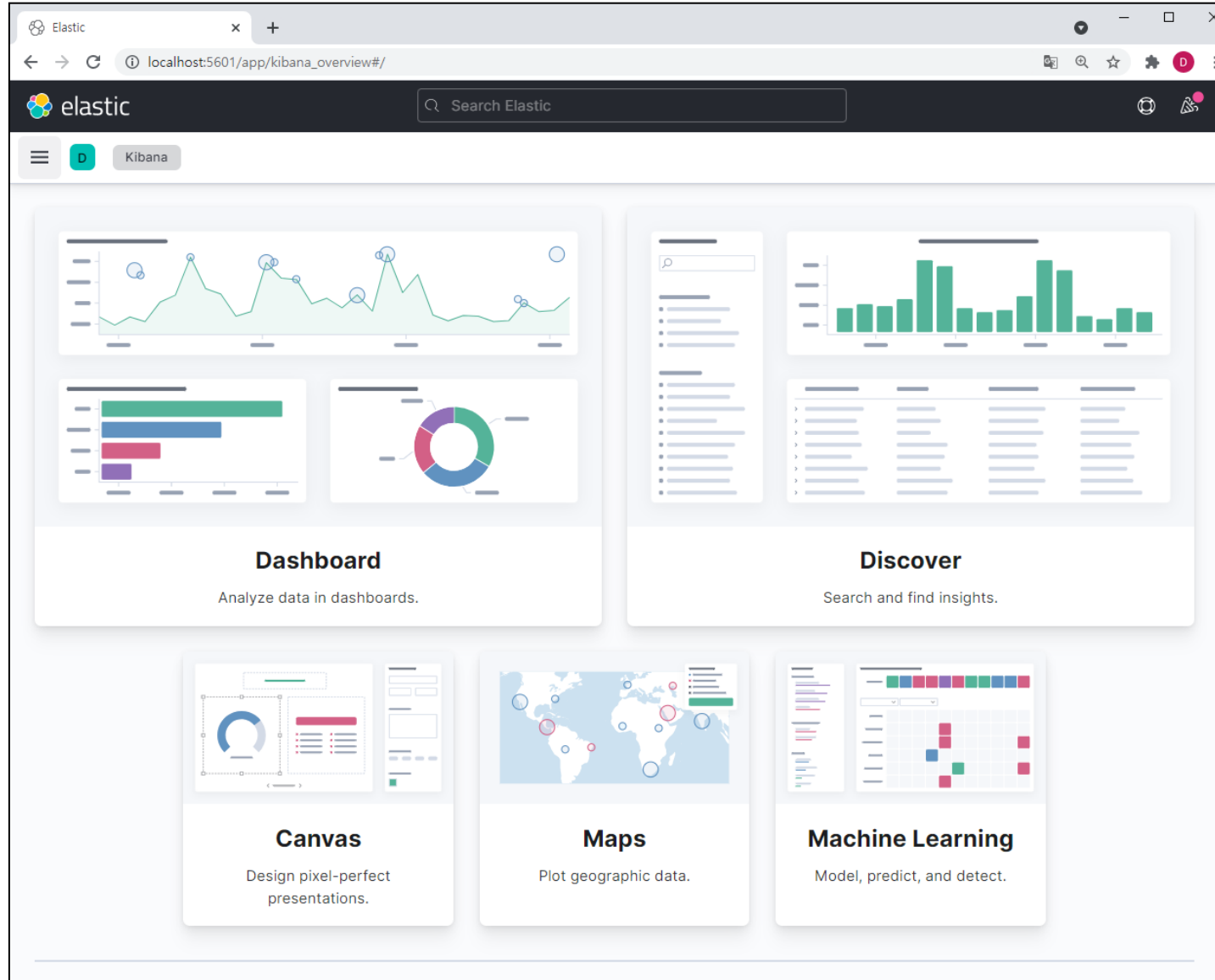
## ElasticSearch.txt



<https://www.elastic.co/kr/webinars/getting-started-elasticsearch>

<https://github.com/kimjmin/elastic-demo/blob/master/demos/get-started/elasticsearch-7x.md>

# Kibana 실습



# ElasticSearch 클러스터 관리도구

<https://chrome.google.com/webstore/detail/elasticsearch-head/ffmkiejjmecolpfloofpjologoblkegm>

The screenshot displays the ElasticSearch Head web interface. At the top, the URL is `http://192.168.7.8:9200/` and the cluster health is **yellow (6, 18)**. The interface includes tabs for Overview, Browser, Structured Query, and Any Request. The main section is titled "Cluster Overview" and lists several nodes: Leon, Pris, Rick, Rachel, Zhora, and Roy. Each node has a status bar with green and red squares indicating shard status. A modal window is open for node "Leon", showing its configuration details.

Node	cu_docs	bnvil	cu_msg	anvil
Leon	size: 180Gb (540Gb), docs: 995131 (995131)	size: 80kb (480kb), docs: 90 (90)	size: 313Gb (1.56Tb), docs: 10047450 (10140915)	index: close
Pris				
Rick				
Rachel				
Zhora				
Roy				
Unassigned				

```
{
  name: "Leon",
  transport_address: "inet[/192.168.7.8:9302]",
  attributes: {},
  http_address: "inet[/192.168.7.8:9202]",
  os: {
    refresh_interval: 3000,
    cpu: {
      vendor: "Intel",
      model: "Macmini4,1",
      mhz: 2400,
      total_cores: 2,
      total_sockets: 1,
      cores_per_socket: 2,
      cache_size: "3kb",
      cache_size_in_bytes: 3072
    }
  }
}
```



# Thank you