# Tổng quan về ElasticSearch

## **ElasticSearch là gì ?**

* Elasticsearch là một **công cụ tìm kiếm và phân tích dữ liệu phân tán, mã nguồn mở**, được phát triển dựa trên **Apache Lucene**. Nó cho phép bạn lưu trữ, tìm kiếm và phân tích lượng lớn dữ liệu một cách nhanh chóng và gần như thời gian thực (near real-time). Elasticsearch cung cấp một giao diện RESTful API đơn giản, dễ sử dụng để tương tác, và hỗ trợ nhiều loại dữ liệu khác nhau
* Elasticsearch như một cơ sở dữ liệu NoSQL chuyên biệt, được tối ưu hóa cho việc tìm kiếm và phân tích dữ liệu

## **Tại sao cần dùng ElasticSearch**

* **Tìm kiếm nhanh và hiệu quả**: Elasticsearch được xây dựng trên nền tảng Lucene, một thư viện tìm kiếm rất mạnh mẽ. Nó sử dụng các kỹ thuật lập chỉ mục (indexing) tiên tiến để cho phép tìm kiếm toàn văn bản (full-text search) và tìm kiếm theo cấu trúc (structured search) với tốc độ cực nhanh, ngay cả với lượng dữ liệu khổng lồ.
* **Khả năng mở rộng**: Elasticsearch được thiết kế để hoạt động theo kiến trúc phân tán. Bạn có thể dễ dàng thêm các node (máy chủ) mới vào cụm Elasticsearch để mở rộng khả năng lưu trữ và xử lý dữ liệu.
* **Phân tích dữ liệu:** Elasticsearch không chỉ là công cụ tìm kiếm mà còn cung cấp các tính năng phân tích dữ liệu mạnh mẽ. Bạn có thể tổng hợp, thống kê, tính toán và trực quan hóa dữ liệu một cách dễ dàng.
* **Tìm kiếm gần thời gian thực (Near Real-Time):** Dữ liệu được lập chỉ mục và có thể tìm kiếm gần như ngay lập tức sau khi được thêm vào.
* **Dễ sử dụng**: Elasticsearch cung cấp RESTful API đơn giản, dễ sử dụng và hỗ trợ nhiều ngôn ngữ lập trình.
* **Linh hoạt:** Elasticsearch có thể lưu trữ và tìm kiếm nhiều loại dữ liệu khác nhau, bao gồm văn bản, số, ngày tháng, địa lý, v.v.
* **Cộng đồng lớn và hỗ trợ mạnh mẽ**: Elasticsearch là một dự án mã nguồn mở với cộng đồng người dùng và nhà phát triển đông đảo, cung cấp nhiều tài liệu, hướng dẫn và hỗ trợ.

## **Các khái niệm cốt lõi trong ElasticSearch**

* **Node:** Một instance (phiên bản) đang chạy của Elasticsearch. Một cụm Elasticsearch bao gồm nhiều node.
* **Cluster:** Một tập hợp các node hoạt động cùng nhau để lưu trữ và xử lý dữ liệu.
* **Index:** Một tập hợp các tài liệu (document) có đặc điểm tương tự nhau. Có thể coi index tương tự như một bảng (table) trong cơ sở dữ liệu quan hệ.
* **Document:** Một đơn vị dữ liệu cơ bản trong Elasticsearch, được biểu diễn dưới dạng JSON. Có thể coi document tương tự như một bản ghi (row) trong cơ sở dữ liệu quan hệ.
* **Type (deprecated in 7.x, removed in 8.x):** Một loại tài liệu trong một index. Trước đây, mỗi index có thể có nhiều type. Hiện tại, mỗi index chỉ nên có duy nhất một type, hoặc không sử dụng type.
* **Field:** Một thuộc tính của document, tương tự như một cột (column) trong cơ sở dữ liệu quan hệ.
* **Mapping:** Định nghĩa cách một document và các field của nó được lưu trữ và lập chỉ mục.
* **Shard:** Một phần của index. Elasticsearch chia nhỏ index thành các shard để phân tán dữ liệu trên nhiều node, tăng khả năng mở rộng và hiệu suất.
* **Replica:** Một bản sao của shard. Replica được sử dụng để tăng tính sẵn sàng và khả năng chịu lỗi.
* **Inverted Index:** Cấu trúc dữ liệu cốt lõi của Elasticsearch (và Lucene), cho phép tìm kiếm toàn văn bản nhanh chóng. Nó ánh xạ các từ (term) đến các tài liệu chứa chúng.

## **Ưu điểm và nhược điểm của ElasticSearch**

* **Ưu điểm:**
  + **Tốc độ tìm kiếm nhanh**: Nhờ vào inverted index và kiến trúc phân tán.
  + **Khả năng mở rộng cao**: Dễ dàng thêm node để tăng khả năng lưu trữ và xử lý.
  + **Tìm kiếm gần thời gian thực:** Dữ liệu được lập chỉ mục và có thể tìm kiếm gần như ngay lập tức.
  + **Phân tích dữ liệu mạnh mẽ**: Cung cấp các tính năng tổng hợp, thống kê và trực quan hóa dữ liệu.
  + **RESTful API:** Dễ sử dụng và tích hợp với các ứng dụng khác.
  + **Hỗ trợ nhiều loại dữ liệu**: Bao gồm văn bản, số, ngày tháng, địa lý, v.v.
  + **Cộng đồng lớn và hỗ trợ mạnh mẽ**: Nhiều tài liệu, hướng dẫn và hỗ trợ từ cộng đồng.
  + **Schema-free**: Không yêu cầu schema cố định.
* **Nhược điểm:**
  + **Độ phức tạp**: Cấu hình và quản trị Elasticsearch có thể phức tạp, đặc biệt là với các cụm lớn.
  + **Tài nguyên**: Elasticsearch có thể tiêu tốn nhiều tài nguyên hệ thống (CPU, RAM, lưu trữ).
  + **Join hạn chế**: Elasticsearch không hỗ trợ các phép join phức tạp như cơ sở dữ liệu quan hệ.
  + **ACID properties**: Elasticsearch không đảm bảo đầy đủ các thuộc tính ACID (Atomicity, Consistency, Isolation, Durability) như cơ sở dữ liệu quan hệ. Nó cung cấp độ bền (durability) thông qua cơ chế replication, nhưng không đảm bảo tính nhất quán (consistency) tuyệt đối trong một số trường hợp.
  + **Bảo mật:** Cần cấu hình cẩn thận để đảm bảo an toàn cho dữ liệu trong Elasticsearch.
  + **Không phải lúc nào cũng là lựa chọn tốt nhất**: Đối với các ứng dụng yêu cầu cập nhật thường xuyên với transaction phức tạp, hoặc cần join dữ liệu từ nhiều bảng, cơ sở dữ liệu quan hệ có thể phù hợp hơn.

# 2.

