**1. Mục đích:**

Dùng để tìm kiếm ttin dạng text thông thường.

Tìm kiếm dạng text có cấu trúc.

Tổng hợp và phân tích dữ liệu KD yc bảo mật.

Xử lý và lưu trữ dữ liệu lượng lớn

Ghi nhật ký và phân tích nhật ký

Kiểm tra hiệu năng hoạt động của ứng dụng.

Tìm kiếm dữ liệu theo tọa độ, tổng quan hóa dữ liệu trên không gian địa lý.  
- Hoạt động như 1 cloud server, tìm kiếm thông qua cơ chế restful (đầu vào và đầu ra là Json.)

Người dùng tạo ra các Http request dạng Json, sau đó nhập vào elastic search. **Các dữ liệu này đã được đánh chỉ mục(index) nên có hiệu suất tifm kiếm cao.**

* Đầu tiên, Dl được đưa vào ES từ nhiều nguồn khác nhau, Dl được phân tích sử lý trong quá trình nhập liệu. Sau đó, DL được phân loại và đánh chỉ mục, và đẩy lên ES, cuối cùng, ng dùng có thể tạo các truy vấn phức tạp và lấy dữ liệu trả về từ ES/

**2. Các thành phần.**

1. **Document** là thông tin cơ bản được lập chỉ mục trong elastic search, là đơn vị lưu trữ nhỏ nhất trong ES, có thể là văn bản, hoặc bất kỳ dạng CTDl nào bằng JSon đã đipwkc mã hóa. Mỗi doc có 1 id duy nhất.

2. **Index (**chỉ mục**)**  là tập hợp các tài liệu có đặc điểm liên quan về mặt logic, vd trong 1 trang web thương mại điện tử, sẽ chỉ tìm thấy 1 chỉ mục cho khách hàng. 1 chỉ mục cho sản phẩm,,… chỉ mục được sử dụng để tìm kiếm, thêm mới, sửa xóa.  
1 khái niệm là  **inverted index**, (chỉ mục đảo ngược), CMĐN không lưu trữ trực tiếp mà chia từng tài liệu thành các cụm từ tìm kiếm riêng lẻ, nhờ đó có thể tìm kiếm nhanh chóng kể cả trong các tệp dữ liệu lớn.

3 **Shard**

Shard là tập con các Document của 1 Index, là đơn vị lưu trữ dữ liệu nhỏ nhất, hoạt động ở mức thấp nhất. Bằng cách phân phối Documents trong một Index trên nhiều Shard, Elasticsearch có thể đảm bảo tính dự phòng, bảo vệ hệ thống khỏi lỗi phần cứng và giúp tăng khả năng truy vấn khi hoạt động.

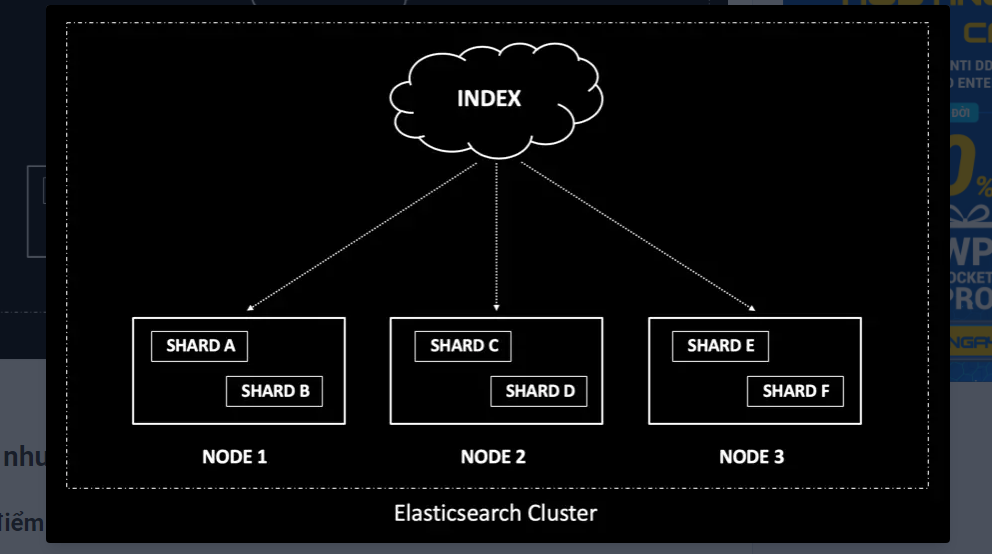
Có 2 loại **shard** : primary shard và replica shard.

Primary shard lưu trữ dl và đánh index, sau đó vận chuyển tới các replica shard.

Replica shard: chứa các dữ liệu copy của primary shard, đảm bảo DL của primary shard được toàn vẹn, ngay cả khi hệ thống có vấn đề.  
4 N**ode**

Là đầu não của elasticsearch, là nơi lưu trữ dữ liệu trực tiếp, đánh index của cluster và thực hiện các thao tác tìm kiếm, 1 node có 1 tên duy nhất.

Cluster là tập hợp các node hoạt động cùng nhau. Mỗi cluster có 1 node chính được chọn tự động. có thể thay thế, 1 trong những lỗi sai của dev là định danh các node trùng tên nhau, sẽ lỗi cho các node. Lưu ý khi setup



Các ưu điểm:  
**Tìm nhanh**: do ES tìm các chỉ mục (index) thay vì tìm các văn bản trực tiếp. , sử dụng cấu trúc document thay vì bảng hay lược đồ.  
**Khả năng mở rộng**: bản chất phân tán (servermaster -nodes) giúp người dùng mở rộng thêm nhiều node, cử lý càng nhiều DL.

**Đơn giản hóa hiển thị và báo cáo DL**: cho phép tích hợp và là BEAT và LOGSTASH, giúp người dùng dễ dàng xử lý dữ liệu trước khi đưa vào Elastic search.

**Tính năng vượt trội**: cho phsp lưu trữ và tìm kiếm hiệu quả như cuộn dữ liệu và quản lý vòng đời index

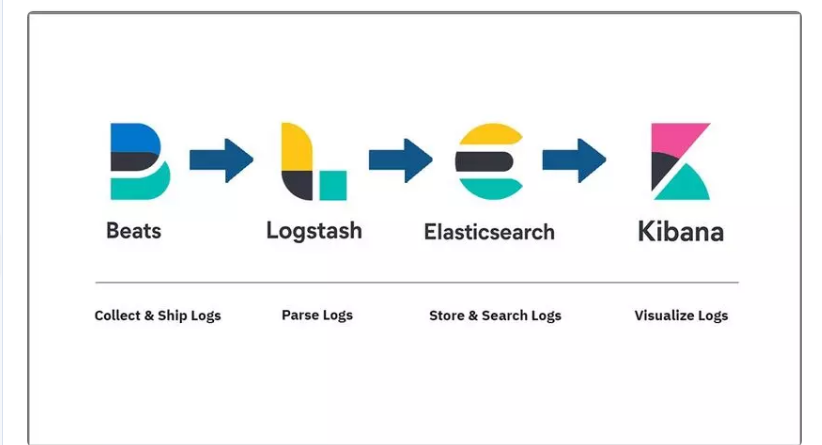
Nhược điểm:

1. thiết kế chủ yếu cho việc search, do vậy không bằng các database nếu sd với mục đích khác.  
2. Không thích hợp với các hệ thống thường xuyên cập nhật dữ liệu liên tục.

Tích hợp các công cụ khác:

- Roles lisst:

+ kibana\_dashboard\_only\_user, apm\_system, watcher\_admin, logstash\_system, rollup\_user, kibana\_user, beats\_admin, remote\_monitoring\_agent, rollup\_admin, code\_user, data\_frame\_transforms\_admin, snapshot\_user, monitoring\_user, logstash\_admin, machine\_learning\_user, data\_frame\_transforms\_user, machine\_learning\_admin, watcher\_user, apm\_user, beats\_system, reporting\_user, kibana\_system, transport\_client, remote\_monitoring\_collector, code\_admin, superuser, ingest\_admin



Sự khác nhau giữ Database thông thường và Elastic search:

| **Elasticsearch** | **RDBMS** |
| --- | --- |
| Index | Table |
| Document | Row |
| Cluster | Database |
| Field | Column |