What is Elasticsearch?

Elasticsearch, by nature, is a distributed document store. This means that it uses complex data structures serialized as JSON documents. With multiple nodes in a cluster, documents are distributed across the cluster and can be accessed immediately from any node.

One of the most significant advantages of using Elasticsearch is the fact that it uses inverted indexes for near real-time full-text searches. This engine makes all data into indexed fields with a dedicated and optimized data structure. Because each data type has its own data structure, it provides the engine with a lot of speed.

Documents in Elasticsearch can be indexed without explicitly specifying how to handle the fields in the document. Dynamic mapping lets Elasticsearch automatically detect and add new fields to the index. This makes it really easy to index and use the data. However, this doesn't mean that you cannot manually map it all because you can explicitly define mappings to control how fields are stored and indexed.

Defining your own mappings enables you to [1]:

- Distinguish between full-text string fields and exact value string fields
- Perform language-specific text analysis
- Optimize fields for partial matching
- Use custom date formats
- Use data types such as geo_point and geo_shape that cannot be automatically detected

With Elasticsearch, you can access the Apache Lucene search engine library. It provides an easy-to-use REST API for managing the cluster, searching, and indexing the data. You can use the developer console in Kibana or the command line to submit requests. Elasticsearch client is available in Java, JavaScript, Go, .NET, PHP, Perl, Python, or Ruby.

Elasticsearch supports structured queries, full text queries, and complex queries with the two combined. You can search for individual terms, phrases, similarity, and prefix searches with autocomplete functionality. This engine also indexes non-textual data in optimized data structures that support high-performance geo and numerical queries. All these search capabilities can be done using Query DSL (a JSON-style query language).

Elasticsearch has the ability to summarize your data and gain insight into key metrics, patters, and trends. This is done by the use of aggregations, which use the same data structures used for search, making them considerably fast. With this, you can analyze and visualize your data in real-time. Also, you can use aggregations alongside search requests, letting you perform analytics and search documents simultaneously, on the same data, with a single request. You can use machine learning features to create accurate baselines of normal behavior in your data and identify anomalous patterns. With machine learning, you can detect[2]:

- Anomalies related to temporal deviations in values, counts, or frequencies
- Statistical rarity
- Unusual behaviors for a member of a population

Elasticsearch is, by nature, distributed, therefore making it able to scale with whatever you might need. You can add nodes or servers to a cluster to increase capacity, and Elasticearch automatically distributes your data and query load across

all of the available nodes. Since it distributes the documents in an index across multiple shards and those shards across multiple nodes, it can ensure redundancy. With two types of shards (primaries and replicas), Elasticsearch can protect against hardware failure and increase capacity to serve read requests like searching or retrieving a document.

"There are a number of performance considerations and trade-offs with respect to shard size and the number of primary shards configured for an index. The more shards, the more overhead there is simply in maintaining those indices. The larger the shard size, the longer it takes to move shards around when Elasticsearch needs to rebalance a cluster." [3]

In case of emergencies like power outages or similar situations, there's CCR (Cross-cluster replication), which provides a way to synchronize primary and secondary cluster indices automatically. Whenever the primary cluster fails, the secondary remote cluster can take over.

References

[1] “Data in: documents and indices | Elasticsearch Guide [8.6] | Elastic,” www.elastic.co. https://www.elastic.co/guide/en/elasticsearch/reference/current/documents-indices.html [2] “Information out: search and analyze | Elasticsearch Guide [8.6] | Elastic,” www.elastic.co. https://www.elastic.co/guide/en/elasticsearch/reference/current/search-analyze.html (accessed Feb. 25, 2023). [3] “Scalability and resilience: clusters, nodes, and shards | Elasticsearch Guide [8.6] | Elastic,” www.elastic.co. https://www.elastic.co/guide/en/elasticsearch/reference/current/scalability.html (accessed Feb. 25, 2023).