**Elastic search:**

**What: (search engine and also can be used to analyze the data and find trend)**

**Fully distributed enterprise grade search and analytics Engine**

**No Sql, distributed, Full Text database**

**Open source**

**It’s based on Lucene engine build on java**

**Accessible through extensive and elaborative restful API**

**Why:**

1. **Implement enterprise search**

**Ex: e-commerce application used to search product**

1. **Analytics –Analyze and find the trend**

**Lot of log with transaction data if you need to find trend you can use it**

**3. Auto Complete and instant Search**

**- Google as soon as you type something it will suggest based on the previous search it will suggest**

1. **Implement Simple, Fuzzy Search ( in google search if you are typing word with Misspell it will try to give Closely matched words)**
2. **Implement Logging & Analytics :**

**You are maintaining server across the globe and you want to analyze the logs of server and see centralized location and se health of the server**

**It will come with 2 more tools**

1. **Logstash 2.Kibana 3.beats utility**

**Communication with search server through HTTP REST API**

**Curl –X <REST verb> <Node>:<Port>/<Index>/<Type>/<ID>**

**Ex: curl –X GET** [**http://localhost:9200/person/employee/123**](http://localhost:9200/person/employee/123)

**Schema –less Json Document ( like Nosql database) 🡪 don’t have define the field and datatype**

**Near real-time search ( if we are updating /delete/insert it will reflect it in the 1 or 2 sec because distributed architecture)**

Fully distributed, if you want to analyze the data transaction or log data or you want to extract useful information, Centralized the logging system where you are capture the different system which is located different place, you want store all the logs and search using elastic search

If we want to create your own search engine,

Collection of different document (RDBMS 🡪Table) **🡪 index (RDBMS 🡪Database)**

Index is split in to multiple Shard (chunk) Shards🡪 logical divide your data so that easily searchable and querable, if we are not shards data searchable will be difficult

Its like our RDBMS (Database)

name should be lowercase,

we can use index when we have update,search,delete document within the index.

You can define many index as you want with in cluster

Type 🡪 represent class/Category of similar document,

You can think of type as a table with in relational database

Consist of name and mapping.

Index can consist of one or more type defined each with their own mapping

Searching for specific document types applies a filter on this field

Mapping 🡪 similar to database schema for table in the RDBMS.

Includes data type for each field, eg.string, integer,date.

Dynamic mapping means that it is optional to define a mapping explicitly

Document 🡪 basic unit of information that can be indexed,

Consist of fields , which are key/ value pairs

Expressed in JSon, you can store as many document with in an index

It’s similar to RDBMS ( Row)

Shards 🡪 can be hosted in one node or multiple nodes, based on the no. of node shard will be distributed .

No.of shards can be specified when creating an index by default is 5

Allows to distribute and parallelize operations across shards,which is increase performance.

Replica 🡪 copy of shard, provide high availability in case a shard or node fails

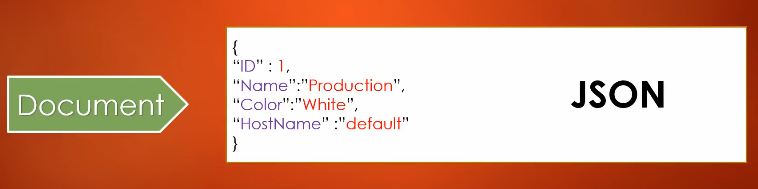
A replica never resides on the same node as the original shard

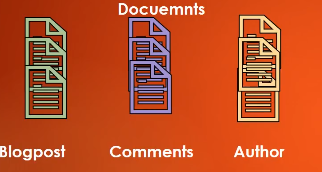
Allows scaling search volume, bcoz search queries can be executed on all replicas in parallel

Cluster 🡪 Collection of Shards (it will be replica of shard since one shard failure we can take it from replica in the cluster), to provide indexing and search capability across all nodes

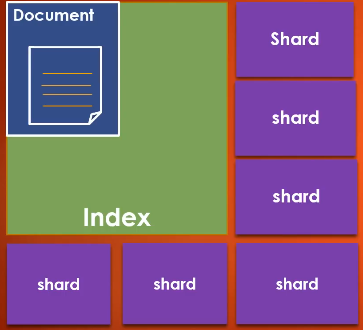
Node 🡪 single server, cluster can contain one node will store all the data

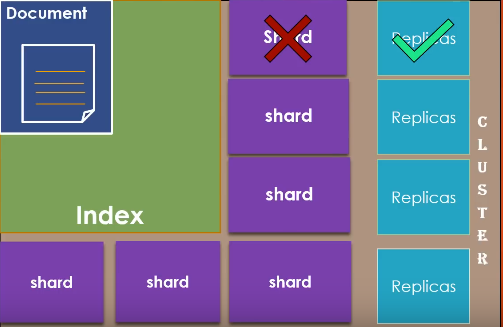
Elastic type doesn’t have array type











Shard distributed into different nodes

