# 5 Most Important Elasticsearch Interview Questions

## **Introduction to Elasticsearch interview questions and answers**

Elasticsearch is a search engine developed by Shay Banon in 2010 based on Apache Lucene project and is cross-platform and was written in [Java Programming Language](https://www.educba.com/java-programming-language-10-things-you-must-know/) having Apache License 2.0. This [search engine](https://www.educba.com/search-engine-marketing-strategies/) is popular and similar to that of Apache Solr which is also a search engine based on Lucene.

Preparing for a job interview in Elasticsearch. I am sure you want to know the most common Elasticsearch interview questions and answers that will help you crack the interview with ease. Below is the list of top 5 Elasticsearch interview questions and answers at your rescue.

Below is the list of Elasticsearch Interview Questions that are mostly asked:

### **1. What is Elasticsearch?**

**Answers:**   
Elasticsearch is a search engine based on Apache Lucene that supports full-text search engine with scheme free [JSON](https://www.educba.com/course/json-training/) objects and HTTP web interface. This is s free and open source project developed in [Java](https://www.educba.com/course/online-java-script-training/) and licensed under Apache License terms. The key components of Elasticsearch are [Node](https://www.educba.com/course/node-js-training/), [Cluster](https://www.educba.com/cluster-analysis-vs-factor-analysis/), Index, Type, Document, Shard and Replicas. Elastic search has the capacity to perform a fast incisive search over large chunks of data.

Elasticsearch can be used to search different kinds of documents that provide scalable search, multi-tenancy, and [real-time](https://www.educba.com/real-time-analytics/) search. Elasticsearch is also available in [Amazon Cloud](https://www.educba.com/course/amazon-cloud-computing-hosting-asp-net-website-amazon-ec2/) as [Amazon Web Services](https://www.educba.com/course/amazon-web-services-aws-cloud-computing-training/) Elasticsearch Cloud. Elasticsearch is a distributed, RESTful search analytics engine that is capable of helping in solving numerous use cases for the business requirement in [big data](https://www.educba.com/big-data-vs-data-warehouse/) or [data science](https://www.educba.com/data-science-vs-software-engineering/) environment.

### **2. What is an index and inverted index in Elasticsearch?**

**Answer:**   
Elasticsearch has a concept called index which is similar to that of a table in a relational [database structure](https://www.educba.com/course/data-structures-and-algorithms-in-java/). An index has mappings that define multiple types. An index maps one or more multiple shards and can have zero or many replica shards. Here Shard is an index which is split into multiple elements. Elasticsearch will have one replica for each index. The main reason for fast searching in elastic search is that index will be searched instead of content making it so faster.

The inverted index is an index which is used to make very fast full-text searches which is a key component. This is used to search and make a list of all unique words searched in all documents. To create the inverted index, first, the field of each document should be split into separate elements. In an inverted index, to store a mapping from content, words or numbers can be used in a database to its file location. The inverted index is a key component and structure of the elastic search to provide very fast full-text searches.

### **3. What is a document in Elasticsearch?**

**Answer:**   
A document in elastic search is a top level or a root component object that is serialized into JSON object and will be stored in Elastic search under a unique id. The entities or objects in most of the applications can be serialized to [JSON](https://www.educba.com/course/web-service-parsing-json-api/) with keys and values where the key is the property or name of the field and value is the data present for that key such as String or Number or Boolean etc.,

Documents in elastic search are indexed and stored and will be available to search using the index. In the key-value pairs, the indexes can be generated using auto-generated id values. Inelastic search, document, and object are often interchangeable words. Mapping is the process to define a document, and the fields it contains which are stored and indexed. In a document, each index will have one mapping type that defines how mapping can be done and a document will be indexed. Each mapping type will have meta fields and fields, where meta fields can be used to customize the document’s metadata. Each field will have data types like Boolean, double, long, date or text etc.

### **4. What is a Node in Elasticsearch?**

**Answer:**   
A node is an important component in an Elasticsearch which is needed before starting an instance of Elasticsearch. A group of nodes is called a cluster. If a single node of [Elasticsearch](https://www.educba.com/hadoop-vs-elasticsearch/) is running, then it is called a cluster of one node. In networking, the transport layer is used to establish communication between nodes of a cluster. Each and every node existing in a cluster can send client requests to each other and can establish communication with each other.

There are several types of nodes such as master node, data node, ingest node and tribe node. A Master node is a node that controls the entire cluster. A data node is a node that holds data in it and performs logical operations on the data. An ingest node is a node that can be used to ingest pipeline which means a series of processors to a document to perform some transformations before indexing the document. A tribe node is a node that performs some coordination to connect to multiple clusters across all the connected clusters and perform some logical operations or searches. By default, a node will always be a master node and a data node, but depending on the large requirements, node configurations should be carried out.

### **5. What is Schema in Elasticsearch?**

**Answer:**   
A schema is a structure that describes multiple fields that provides the detailed overview of the document and its type and the way of handling the fields inside the document. The schema is used for mapping in Elasticsearch which describes the fields in JSON documents with its data types. This process is called schema mapping in Elasticsearch. An Elasticsearch server usually contains zero or more indexes. An index contains multiple types which will have multiple documents in them. The other feature of elastic search is that it can also be schema-less by making the documents to be indexed without providing schema clearly.

If a mapping is not explicitly provided in elastic search, then a default mapping will be generated automatically while detecting fields during the process of indexing. This is the process of dynamic mapping generation. The mapping will be done in the form of JSON in elastic search and this will be the hierarchically structured format. Each level in the hierarchy will be having properties configuration to make it work flexibly as per requirement. This means each and every level and its child levels will be having each property set to the last level.

## **ElasticSearch Interview Questions and Answers**

**What is ElasticSearch?**

Elasticsearch is a search engine based on Lucene. It has a distributed, multitenant-able full-text search engine. Elasticsearch is developed in Java and is released as open source under the terms of the Apache License.

**What is the use of attributes- enabled, index and store?**

The enabled attribute is applicable to several ElasticSearch created fields like \_index and \_size.

Store implies the data stored by Lucene, which will again return when necessary. Stored fields are not searchable.

The index is employed for searching. Indexed fields are transformed during analysis, and cannot retrieve the original data when necessary.

**What is an Analyzer in ElasticSearch?**

While indexing data, it is transformed internally via the defined Analyzer for the index.

Analyzers are made of one Tokenizer, preceded by CharFilters and zero or many TokenFilters. On the other hand, analysis module refers Analyzers under the name of mapping definitions or any APIs.

Elasticsearch is prebuilt with analyzers that are ready to use. However, you can integrate the built in character, token filters, along with tokenizers to create custom analyzers.

**What is Character Filter in Elasticsearch Analyzer?**

A character filter obtains the ideal text as stream of characters, later on modifies it by adding, deleting, or altering characters. For example, any character filter in usage has the ability to convert Hindu-Arabic numerals (٠‎١٢٣٤٥٦٧٨‎٩‎) into Arabic-Latin numerals (0123456789), and even sometimes strip [HTML elements](https://tekslate.com/html-formatting-elements-tags/) via the stream.

**What is Token filters in Elasticsearch Analyzer?**

A token filter obtains the token stream, later on add, delete, or alter the tokens. For instance, a lowercase token filter modifies all tokens into lowercase, a stop token filter deletes stop words, and a synonym token filter includes synonyms into the token stream.

Token filters will be unable to change the position or character offsets of any certain token.

**What is a Tokenizer?**

Tokenizers break down a string into stream of tokens. A single tokenizer split the string into terms when working with punctuation and whitespace. Elasticsearch has a number of built in tokenizers which can be used to build custom analyzers.

**What is a Filter?**

After Tokenizer ends the process of data, the same is carried by Filter.

Certain types of Filters available in ElasticSearch 1.10, are.

* AND FILTER
* EXISTS FILTER
* GEO DISTANCE FILTER
* GEO POLYGON FILTER
* GEOHASH CELL FILTER
* HAS PARENT FILTER
* INDICES FILTER
* MATCH ALL FILTER
* NESTED FILTER
* OR FILTER
* QUERY FILTER
* REGEXP FILTER
* TERM FILTER
* TYPE FILTER

**What are the advantages of Elasticsearch?**

* Elasticsearch is compatible on any platform.
* Elasticsearch is Near Real Time (NRT), making it searchable on engine.
* Elasticsearch cluster is distributed, scalable and easy to integrate.
* Elasticsearch REST uses JSON objects, making it to invoke the Elasticsearch server along with different programming languages.
* Elasticsearch supports every document type except text rendering.

**What is Elasticsearch REST API and use of it?**

Elasticsearch provides a very comprehensive and powerful REST API that you can use to interact with your cluster. Among the few things that can be done with the API are as follows:

1. Check your cluster, node, and index health, status, and statistics
2. Administer your cluster, node, and index data and metadata
3. Perform CRUD (Create, Read, Update, and Delete) and search operations against your indexes
4. Execute advanced search operations viz. aggregations, filtering, paging, scripting, sorting, among many others

**What are the Disadvantages of Elasticsearch?**

* Elasticsearch does not support multiple languages while handling request and response data in JSON.
* In rare cases, it has a problem of Split Brain situations.

**Does ElasticSearch have a schema?**

Yes, Elasticsearch can have a schema. A schema is a description of one or more fields that describes the document type and how to handle the different fields of a document. The schema in Elasticsearch is a mapping that emphasizes the JSON document fields and other data type, as well as Lucene indexes under the hood. Because of this, in Elasticsearch terms, we usually call this schema a “mapping”.  
  
**What is a cluster in ElasticSearch?**

Cluster is a collection of nodes that holds data together and enables indexing and search abilities across each. Each cluster is recognized by a unique default name i.e. “elasticsearch”. This name is important because a node can only be part of a cluster if the node is set up to join the cluster by its name.

**What is a node in ElasticSearch?**

Node is a minute server and forms a part of the cluster. It stores the data and enjoys the clusters indexing and search functionalities.

**What is Ingest Node in Elasticsearch?**

Ingest nodes can execute pre-processing an ingest pipeline. It effectively transform and works on the document prior to indexing. Dedicated ingest nodes mark the master and data nodes either as false or true.

**What is Elasticsearch Data Node?**

Data nodes hold shards that handle indexed documents. They execute data related CRUD and search aggregation operations etc. Set node.data=true to make node as Data Node.

Data Node operations are I/O-, memory-, and CPU-intensive. Data nodes benefit the separation of the master and data roles.

**What is Master Node and Master Eligible Node in Elasticsearch?**

Master Node control cluster wide operations like to create or remove an index, track nodes of cluster, and decide to allocate shards on nodes. It is important for cluster health to have a stable master node. Master Node elected based on configuration properties node.master=true (Default).

Master Eligible Node decide based on below configuration

discovery.zen.minimum\_master\_node: number (default 1)

And above number decide based (master\_eligible\_nodes / 2) + 1

**What is Tribe Node and Coordinating Node in Elasticsearch?**

Tribe node connect variant clusters and execute search operations across each connected clusters. This node is configured by settings *tribe*.

Coordinating Node is just like a Smart Load balancer that handle master duties, to hold data, and pre-process documents, then you are left with a coordinating node that can only route requests, handle the search reduce phase, and distribute bulk indexing.

Every node can be termed as a coordinating node which has all three node.data, node.ingest and node.master, set to false. This node is impossible to disable as it possess enough memory and CPU to deal with the gather phase.

**What is an index in ElasticSearch?**

Index is a ‘database’ within relational database. Its mapping defines multiple types and maps to one or many primary shards and can have zero or many replica shards.

MySQL => Databases

ElasticSearch => Indices

**What is inverted index in Elasticsearch?**

Inverted Index is backbone of Elasticsearch which make full-text search fast.  Inverted index consists of a list of all unique words that occurs in documents and for each word, maintain a list of documents number and positions in which it appears.

For Example: There are two documents and having content as:

1: FacingIssuesOnIT is for ELK.

2: If ELK check FacingIssuesOnIT.

To make inverted index each document will split in words (also called as terms or token) and create below sorted index.

|  |  |  |
| --- | --- | --- |
| **Term** | **Doc\_1** | **Doc\_2** |
| FacingIssuesOnIT | X | X |
| Is | X |  |
| For | X |  |
| ELK | X | X |
| If |  | X |
| Check |  | X |

Now when we do some full-text search for String will sort documents based on existence and occurrence of matching counts.

Usually in Books we have inverted indexes on last pages. Based on the word we can thus find the page on which the word exists.

**What is a shard?**

Different applications need to employ multiple ElasticSearch instances on separate machines. Data in every index is divided into multiple partitions, each controlled by a separate ElasticSearch instance. Each such partition is termed as shard. By default, each ElasticSearch index possess 5 shards.

**What is a replica?**

Each shard has 2 copies called replicas. They are highly-available and fault-tolerant.

**What is a document in ElasticSearch?**

Document is similar to a row in relational databases. Each document in the index possess different structure, but has same data type for mutual fields.

MySQL => Databases => Tables => Columns/Rows

ElasticSearch => Indices => Types => Documents with Properties

**What are the basic operations you can perform on a document?**

The following operations can be performed on documents

1. INDEXING A DOCUMENT USING ELASTICSEARCH.
2. FETCHING DOCUMENTS USING ELASTICSEARCH.
3. UPDATING DOCUMENTS USING ELASTICSEARCH.
4. DELETING DOCUMENTS USING ELASTICSEARCH.

**What is a type in ElasticSearch?**

Type is a logical index partition whose semantics are entirely upon the user.

**What are common area of use Elasticsearch?**

It’s useful in application where need to do analysis, statics and need to find out anomalies on data based on pattern.

It’s useful where need to send alerts when particular condition matched like stock market, exception from logs etc.

It’s useful with application where log analysis and issue solution provide because of full search in billions of records in milliseconds.

It’s compatible with application like Filebeat, Logstash and Kibana for storage of high volume data for analysis and visualize in form of chart and dashboards.

# [ElasticSearch interview questions](https://www.onlineinterviewquestions.com/elasticsearch-interview-questions/)

**1. What is Elasticsearch?**

Elasticsearch is a search engine that is based on Lucene. It offers a distributed, multitenant – capable full-text search engine with as HTTP (Hyper Text Transfer Protocol) web interface and Schema-free [JSON](https://www.onlineinterviewquestions.com/json-interview-questions/)(JavaScript Object Notation) documents. It is developed in Java and is an open source released under Apache License.

**2. What is a current stable version of Elasticsearch?**

As on March 2018, the version 6.2.2 is the latest and stable version of Elasticsearch.

**3. What is Elasticsearch?**

Elasticsearch is a search engine that is based on Lucene. It offers a distributed, multitenant – capable full-text search engine with as HTTP (Hyper Text Transfer Protocol) web interface and Schema-free [JSON](https://www.onlineinterviewquestions.com/json-interview-questions/)(JavaScript Object Notation) documents. It is developed in Java and is an open source released under Apache License.

**4. What is a current stable version of Elasticsearch?**

As on March 2018, the version 6.2.2 is the latest and stable version of Elasticsearch.

**5. Can you list some companies that use Elasticsearch?**

**Some of the companies that use Elasticsearch along with Logstash and Kibana are:**

* Wikipedia
* Netflix
* Accenture
* Stack Overflow
* Fujitsu
* Tripwire
* Medium
* Swat.io
* Hip chat
* IFTTT

**6. What is a Cluster in Elasticsearch?**

It is a set or a collection of one or more than one nodes or servers that hold your complete data and offers federated indexing and search capabilities across all the nodes. It is identified by a different and unique name that is “Elasticsearch” by default.  
This name is considered to be important because a node can be a part of a cluster only if it is set up to join the cluster by its name.

**7. What is a Node?**

Each and every instance of Elasticsearch is a node. And, a collection of multiple nodes which can work in harmony form an Elasticsearch cluster.

**8. What is an Index?**

An index in Elasticsearch is similar to a table in relational databases. The only difference lies in storing the actual values in the relational database, whereas that is optional in Elasticsearch.  
An index is capable of storing actual or analyzed values in an index.

**9. What is a type of Elastic search?**

A type in Elasticsearch is a logical category of the index whose semantics are completely up to the user.

**10. Please Explain Mapping?**

Mapping is a process which defines how a document is mapped to the search engine, searchable characteristics are included such as which fields are tokenized as well as searchable.  
In Elasticsearch an index created may contain documents of all “mapping types”.

**11. What is Document?**

A document in Elasticsearch is similar to a row in relational databases. The only difference is that every document in an index can have a different structure or fields but having the same data type for common fields is mandatory. Each field with different data types can occur multiple times in a document.  
The fields can also contain other documents.

**12. What are SHARDS?**

There are resource limitations like RAM, vCPU etc., for scale out, due to which applications employ multiple instances of Elasticsearch on separate machines.  
Data in an index can be partitioned into multiple portions which are managed by a separate node or instance of Elasticsearch. Each such portion is called a Shard. And an Elasticsearch index has 5 shards by default.

**13. What is REPLICAS?**

Each shard in elastic search has again two copies of the shard that are called the replicas.  
They serve the purpose of fault tolerance and high availability.

**14. How to add or create an index in Elastic Search Cluster?**

By using the command PUT before the index name, creates the index and if you want to add another index then use the command POST before the index name.  
**Ex:** PUT website

An index named computer is created

**15. How to delete an index in Elastic search?**

To delete an index in Elasticsearch use the command DELETE /index name.

**Ex:** DELETE /website

**16. How to list all indexes of a Cluster in ES. ?**

By using GET / \_index name/ indices we can get the list of indices present in the cluster.

**17. How to add a Mapping in an Index?**

Basically, Elasticsearch will automatically create the mapping according to the data provided by the user in the request body. Its bulk functionality can be used to add more than one JSON object in the index.

**Ex:** POST website /\_bulk

**18. How can you retrieve a document by ID in ES. ?**

To retrieve a document in Elasticsearch, we use the GET verb followed by the \_index, \_type, \_id.  
**Ex:** GET / computer / blog / 123?=pretty

**19. How relevancy and scoring is done in Elasticsearch?**

The Boolean model is used by the Lucene to find the similar documents, and a formula called practical scoring function is used to calculate the relevance.  
This formula copies concepts from the inverse document/term-document frequency and the vector space model and adds the modern features like coordination factor, field length normalization as well.  
Score (q, d) is the relevance score of document “d” for query “q”.

**20. What are different ways of searching in Elasticsearch?**

We can perform the following searches in Elasticsearch:

* **Multi-index, Multitype search:**All search APIs can be applied across all multiple indices with the support for the multi-index system.  
  We can search certain tags across all indices as well as all across all indices and all types.
* **URI search:**A search request is executed purely using a URI by providing request parameters.
* **Request body search:** A search request can be executed by a search DSL, that includes the query DSL within the body.

**21. List different types of queries supported by Elasticsearch?**

The Queries are divided into two types with multiple queries categorized under them.

* **Full-text queries:** Match Query, Match phrase Query, Multi match Query, Match phrase prefix Query, common terms Query, Query string Query, simple Query String Query.
* **Term level queries:** term Query, term set Query, terms Query, Range Query, Prefix Query, wildcard Query, regexp Query, fuzzy Query, exists Query, type Query, ids Query.

**22. What is the difference between Term-based and Full-text queries?**

* **Term-based Queries:**Queries like the term query or fuzzy query are the low-level queries that do not have analysis phase. A term Query for the term Foo searches for the exact term in the inverted index and calculates the IDF/TF relevance score for every document that has a term.
* **Full-text Queries:**Queries like match query or query string queries are the high-level queries that understand that mapping of a field. As soon as the query assembles the complete list of items it executes the appropriate low-level query for every term, and finally combines their results to produce the relevance score of every document.

**23. How does aggregation work in Elasticsearch?**

The aggregation framework provides aggregated data based on search query. It can be seen as a unit of work that builds analytic information over the set of documents. There are different types of aggregations with different purpose and outputs.

**24. Where is Elasticsearch data stored?**

Elasticsearch is a distributed documented store with several directories. It can store and retrieve the complex data structures that are serialized as JSON documents in real time.

**25. Can Elasticsearch replace database?**

Yes, Elasticsearch can be used as a replacement for a database as the Elasticsearch is very powerful.  
It offers features like multitenancy, sharding and Replication, distribution and cloud Real-time get, Refresh, commit, versioning and re-indexing and many more, which make it an apt replacement of a database.

**26. How to check elastic search server is running?**

Generally, Elasticsearch uses the port range of 9200-9300.  
So, to check if it is running on your server just type the URL of the homepage followed by the port number.

**Ex:** mysitename.com:9200

**ElasticSearch Fundamentals – Revision Notes**

* Each Elasticsearch shard is a Lucene index
* The number of shards and replicas can be defined per index at the time of creation of the index. The number of replicas per shard can later be changed.
* Shard in ElasticSearch is primarily a Lucene index made up of one or more Lucene segments which store the document data in form of an inverted index.
* Lucene segments are immutable
* Average shard size could vary from 10GB to 40 GB depending upon the nature of data stored in the index. It is commonly seen that time-based data is stored in shard size of 20-40 GB.
* It is recommended to run force-merge operation of merging multiple smaller segments into a larger one in off-peak hours (when no more data is written to the index).
* It is recommended to have 20-25 shards per GB heap space. Thus, a node with 20 GB heap can have 400-500 shards.
* Each shard has metadata related to shard and segment which needs to be stored in memory, and thus, use heap space.
* The size of the shard could be managed based on one of the following techniques:
  + Creating shards based on time-based indexing
  + Creating shards based on documents count for each shard and using rollover API
  + Merging/shrinking existing shards into new shard using Shrink APIs
* It is recommended to determine the maximum shard size from a query performance perspective based on the benchmark using realistic data and queries. There is no thumb rule or one-size-fits-all solution to this.
* It is recommended to use time-based indices for managing data retention whenever possible. Data can be grouped into indices based on the retention period. This makes it manage the indices in terms of creating and deleting the indices.

**Sample Interview Questions**

* Explain the concepts of the cluster, node, index, shard, and replicas?
* How to determine the shard size? What is recommended as the size of shard consisting of time-based data?
* How does update and delete documents from index works?
* How many shards can be allocated to a node having the memory of 20 GB or so?
* Explain Lucene segments and merging of segments?
* What is rollover and shrink APIs used for?

**Sample Quiz (Objective Questions) on ElasticSearch**

**How many shards are created by default when elasticsearch server starts?**

2

3

4

5

**Correct! 5**

**How many replicas are created by default for each shard?**

1

2

3

4

**Correct! 1**

**How many shards including primary and replica shards in total are created by default?**

4

6

8

10

**Correct! 10**

**Shards can further be splitted into multiple shards**

True

False

**Correct! False**

**Number of shards of an index can be changed at any point of time**

True

False

**Correct! False**

**Data is available for querying as soon as \_\_\_\_\_\_\_**

It is written to a shard

After the shard is published to the Lucene segment disk

**Correct! After the shard is published to the Lucene segment disk**

**Lucene segments are immutable**

True

False

**Correct! True**

**Updating a document results in which of the following**

Updating the original document in real time

Finding the matching document, marking the document as deleted and adding the new version

**Correct!** Finding the matching document, marking the document as deleted and adding the new version

**Deleting the document results in which of the following**

Deleting the document in the index in real time

Finding the matching document, marking it as deleted.

**Correct!** Finding the matching document, marking it as deleted.

**The more heap space a node has, the more data and shards it can handle.**

True

False

**Correct! True**

**Number of shards on a node depends upon the available heap space**

True

False

**Correct! True**

**Smaller the shard size, smaller is the segment, greater is the overhead**

True

False

**Correct! True**

**Each query is executed in a single thread per shard**

True

False

**Correct! True**

**Which of the following API are used to create a new index given a pre-defined count of documents to be stored in an index is reached?**

Shrink

Rollover

**Correct! Rollover**

**Which of the following API is used to reduce the number of shards in case many shards have been configured initially**

Shrink

Rollover

**Correct! Shrink**

**Creating multiple shards of an index and portioning the data into different indices are one and the same thing**

True

False

**Correct! False**

1. **Question 1. What Is An Index In Elasticsearch?**

**Answer:**

An index is similar to a table in relational databases. The difference is that relational databases would store actual values, which is optional in ElasticSearch. An index can store actual and/or analyzed values in an index.

1. **Question 2. What Is A Document In Elasticsearch?**

**Answer:**

A document is similar to a row in relational databases. The difference is that each document in an index can have a different structure (fields), but should have same data type for common fields.

Each field can occur multiple times in a document with different data types. Fields can contain other documents too.

1. **Question 3. Does Elasticsearch Have A Schema?**

**Answer:**

Yes, ElasticSearch can have mappings which can be used to enforce schema on documents.

1. **Question 4. What Is A Document Type In Elasticsearch?**

**Answer:**

A document type can be seen as the document schema / mapping definition, which has the mapping of all the fields in the document along with its data types.

1. **Question 5. What Is Indexing In Elasticsearch?**

**Answer:**

The process of storing data in an index is called indexing in ElasticSearch. Data in ElasticSearch can be dividend into write-once and read-many segments. Whenever an update is attempted, a new version of the document is written to the index.

1. **Question 6. What Is A Node In Elasticsearch?**

**Answer:**

Each instance of ElasticSearch is called a node. Multiple nodes can work in harmony to form an ElasticSearch Cluster.

1. **Question 7. What Is A Shard In Elasticsearch?**

**Answer:**

Due to resource limitations like RAM, vCPU etc., for scale-out, applications need to employ multiple instances of ElasticSearch on separate machines. Data in an index can be divided into multiple partitions, each handled by a separate node (instance) of ElasticSearch. Each such partition is called a shard. By default an ElasticSearch index has 5 shards.

1. **Question 8. What Is A Replica In Elasticsearch?**

**Answer:**

Each shard in ElasticSearch has 2 copy of the shard. These copies are called replicas. They serve the purpose of high-availability and fault-tolerance.

1. **Question 9. What Is An Analyzer In Elasticsearch?**

**Answer:**

While indexing data in ElasticSearch, data is transformed internally by the Analyzer defined for the index, and then indexed. An analyzer is built of tokenizer and filters. Following types of Analyzers are available in ElasticSearch 1.10.

* + STANDARD ANALYZER
  + SIMPLE ANALYZER
  + WHITESPACE ANALYZER
  + STOP ANALYZER
  + KEYWORD ANALYZER
  + PATTERN ANALYZER
  + LANGUAGE ANALYZERS
  + SNOWBALL ANALYZER
  + CUSTOM ANALYZER

1. **Question 10. What Is A Tokenizer In Elasticsearch?**

**Answer:**

A Tokenizer breakdown fields values of a document into a stream, and inverted indexes are created and updates using these values, and these stream of values are stored in the document.

1. **Question 11. What Is A Filter In Elasticsearch?**

**Answer:**

After data is processed by Tokenizer, the same is processed by Filter, before indexing. Following types of Filters are available in ElasticSearch 1.10.

* + AND FILTER
  + BOOL FILTER
  + EXISTS FILTER
  + GEO BOUNDING BOX FILTER
  + GEO DISTANCE FILTER
  + GEO DISTANCE RANGE FILTER
  + GEO POLYGON FILTER
  + GEOSHAPE FILTER
  + GEOHASH CELL FILTER
  + HAS CHILD FILTER
  + HAS PARENT FILTER
  + IDS FILTER
  + INDICES FILTER
  + LIMIT FILTER
  + MATCH ALL FILTER
  + MISSING FILTER
  + NESTED FILTER
  + NOT FILTER
  + OR FILTER
  + PREFIX FILTER
  + QUERY FILTER
  + RANGE FILTER
  + REGEXP FILTER
  + SCRIPT FILTER
  + TERM FILTER
  + TERMS FILTER
  + TYPE FILTER

1. **Question 12. What Is The Query Language Of Elasticsearch?**

**Answer:**

ElasticSearch uses the Apache Lucene query language, which is called Query DSL.

1. **Question 13. What Is Elasticsearch?**

**Answer:**

Elasticsearch is a search engine based on Lucene. It provides a distributed, multitenant-capable full-text search engine with an HTTP web interface and schema-free JSON documents. Elasticsearch is developed in Java and is released as open source under the terms of the Apache License.

1. **Question 14. What Are The Basic Operations You Can Perform On A Document?**

**Answer:**

The following operations can be performed on documents

* + INDEXING A DOCUMENT USING ELASTICSEARCH.
  + FETCHING DOCUMENTS USING ELASTICSEARCH.
  + UPDATING DOCUMENTS USING ELASTICSEARCH.
  + DELETING DOCUMENTS USING ELASTICSEARCH.

Perform basic operations with Elasticsearch.

1. **Question 15. What Is Inverted Index In Elasticsearch?**

**Answer:**

Inverted index is the heart of search engines. The primary goal of a search engine is to provide speedy searches while finding the documents in which our search terms occur. Inverted index is a hash map like data structure that directs users from a word to a document or a web page. It is the heart of search engines. Its main goal is to provide quick searches for finding data from millions of documents.

Usually in Books we have inverted indexes as below. Based on the word we can thus find the page on which the word exists.

Consider the following statements

* + javainuse is a good website
  + Javainuse is one of the good websites.

For indexing purpose the above text are tokenized into separate terms and all the unique terms are stored inside the index with information such as in which document this term appears and what is the term position in that document.

So the inverted index for the document text will be as follows-

When you search for the term website OR websites, the query is executed against the inverted index and the terms are looked out for, and the documents where these terms appear are quickly identified.

1. **Question 16. What Is A Cluster In Elasticsearch?**

**Answer:**

Cluster is a collection of one or more nodes (servers) that together holds your entire data and provides federated indexing and search capabilities across all nodes. A cluster is identified by a unique name which by default is "elasticsearch". This name is important because a node can only be part of a cluster if the node is set up to join the cluster by its name.