InterviewBit

ElasticSearch Interview Questions

To view the live version of the page, click here.

Contents

ElasticSearch Interview Questions for Freshers

ElasticSearch Interview Questions for Experienced

ElasticSearch Interview Questions for Experienced

(.....Continued)

19. What is an Analyzer ElasticSearch?
20. What is an Inverted index in ElasticSearch?
21. Describe the functionality of the cat API in ElasticSearch.
22. What are the different ElasticSearch commands available in the cat API?
23. Explain ELK stack and its architecture.
24. What configuration management tools does Elasticsearch support?
25. Is it necessary to install X-Pack for Elasticsearch? What are some essential X-pack commands?
26. What do you mean by aggregation in ElasticSearch?
27. Does ElasticSearch have a schema?
28. How can we perform a search in Elasticsearch?
29. Explain Query DSL in ElasticSearch.
30. What types of queries does ElasticSearch support?

Let's get Started

Among the most important aspects of modern software applications are search and analytics. In many applications - from mobile devices to web applications to data analytics tools -scalability and the ability to deal with large volumes of data in near real-time are key requirements. In the modern age, features such as autocomplete in search fields, search suggestions, and location search are widely accepted. This is where Elasticsearch comes into play since it is often the search engine that powers such experiences. As one of the most popular and powerful search engines, Elasticsearch, when combined with Beats, Logstash, and Kibana, can provide integrated solutions to any query. It is known for its distributed nature, simple REST APIs, speed, and scalability.

Have you been looking for a job that requires knowledge of Elasticsearch and the wider Elastic Stack? If so, this is the perfect opportunity for you. Here we have compiled an exhaustive list of top ElasticSearch Interview Questions and answers for both Freshers and Experienced that will help you attain a dream career as an ElasticSearch Engineer.

Before we begin, let's have a quick look at what ElasticSearch is all about.

What is ElasticSearch?

Elasticsearch is a modern, distributed, and analytics search engine that is based or built on Apache Lucene. ElasticSearch enables you to store, search, and analyze vast or huge amounts of data in near real-time, providing results in milliseconds. Elasticsearch, one of the pillars of the Elastic Stack, is a free and open collection of tools for ingesting, storing, enriching, analyzing, and visualizing data. The latency between the time a document is indexed and the moment it can be searched is extremely short with ElasticSearch - typically one second. As opposed to most NoSQL databases, Elasticsearch NoSQL focuses more on search capabilities and provides a rich HTTP RESTful API that allows for fast searches in near real-time. Developed in Java, ElasticSearch is an open-source search engine that has been used by many large organizations around the globe since it was released in 2010.

Now let's look at the most common interview questions you might be asked in your Elasticsearch Technical Interview.

ElasticSearch Interview Questions for Freshers

1. What are the advantages of ElasticSearch?

ElasticSearch's advantages can be listed or summarized as follows:

- **Elasticsearch is a fast search engine:** Since Elasticsearch is built on top of Apache Lucene, it provides a full-text search. The latency between the time a document is indexed and the moment it can be searched is extremely short with ElasticSearch - typically one second. This makes Elasticsearch a good choice for time-sensitive use cases such as infrastructure monitoring and security analytics.
- **Elasticsearch is a distributed search engine:** Elasticsearch stores or distributes documents across several containers called shards, which are duplicated to provide redundant copies of the data in the event of a hardware or system failure. Due to Elasticsearch's distributed nature, it can scale up to thousands of servers and can handle petabytes of data. You can use ElasticSearch as a replacement for document stores such as RavenDB and MongoDB.
- **Elasticsearch provides a wide range of features**: Aside from being scalable, fast, and resilient, Elasticsearch offers numerous built-in features like data roll-ups and index lifecycle management that make storing and searching data easier and more efficient.
- **Data ingestion, visualization, and reporting are simplified with the Elastic Stack or BELEK:** Data can be collected and processed easily using Beats and Logstash before being indexed in Elasticsearch. Besides providing real-time visualization of Elasticsearch data, Kibana provides UIs for quick access to log files, application performance monitoring (APM), and infrastructure metrics data.

2. What is ElasticSearch used for?

Elasticsearch's speed and scalability as well as its ability to index different types of data make it ideal for a number of use cases. In addition to its high scalability, Elasticsearch also offers near-real-time search capabilities. All this adds up to a solution that offers much more than a search engine and supports many operational and critical business use cases. Since Elasticsearch has powerful search capabilities, it is typically the underlying technology for applications requiring complex search requirements. Listed below are some of the use cases of ElasticSearch:

- Application search, Enterprise search, and Website search.
- Analyzing log data in near-real-time and on a scalable basis.
- Business analytics and security analytics.
- Analysis and visualization of geospatial data.
- Monitoring the performance of applications.
- Monitoring infrastructure metrics and containers.

3. How does ElasticSearch work?

The Working of ElasticSearch is summarized as follows:

- Firstly, raw data is gathered from a variety of sources, such as log files, system metrics, or web applications. Beats are lightweight data shipping agents (data shippers) that collect different types of data and forward it to Logstash.
- This raw data is normalized, analyzed, and enriched prior to being indexed in Elasticsearch. Logstash performs several transformations and enhancements, and then sends the data to be indexed in Elasticsearch.
- As soon as the data has been indexed in Elasticsearch, then users can run queries against it, and then aggregate it to generate insights. ElasticSearch enables you to store, search, and analyze vast or huge amounts of data in near real-time, providing results in milliseconds.
- Lastly, from Kibana, users can create powerful visualizations of data, and visualize complex queries through interactive diagrams, geospatial data, and graphs.

4. Can you please list out different ElasticSearch data types for the document fields?

Field types (also called field data types) describe the type of information or data a field contains, such as a string or boolean, and its intended use. The following are some data types for document fields:

Common data types:

- **Binary:** A binary value that is encoded as a Base64 string.
- **Boolean:** A true or false value.
- **Keywords:** The keyword family, which includes the keyword, constant keyword, and wildcard.
- **Numbers:** Numeric types such as long, double, float, bytes, integer, etc.
- **Dates:** Date types, such as date_nano, date.
- **Alias**: Represents the alias of an existing field.

Objects and relational types:

- **Object:** Represent a JSON object.
- **Nested:** A JSON object that maintains a relationship between its subfields.
- **Flattened:** An entire JSON object represented by a single field value.\
- **Join:** Establishes a parent/child relationship between documents within an index.

Structured and Spatial data types:

- **Range:** Range types, like date_range, long_range, float_range, double_range, and IP_range.
- **Point:** Arbitrary cartesian points.
- **Geo_point:** Longitude and latitude points
- **Shape:** Arbitrary cartesian geometries.
- **Geo_shape:** Complex shapes like polygons.

5. How do you stop the ElasticSearch search service from running on a Linux server?

To shut down or turn off the Elasticsearch service on a Linux server, you will need to 'kill' the running process. It is accomplished by sending a SIGTERM request to the process, which ends or terminates it.

In order to initiate the shutdown process, you must first determine the process identifier (PID) for the Elasticsearch service you wish to terminate. Grep command can be used to locate processes easily. If you wish to locate all Elasticsearch-related processes running on a server, you can use the following command:

```
ps -ef | grep elas
```

After identifying the correct PID, simply execute a kill command with the PID of the Elasticsearch process. Upon successful execution of the kill command, Elasticsearch should no longer be running.

6. What is ElasticSearch Mapping?

ElasticSearch mappings define how documents and their fields are indexed and stored in ElasticSearch databases or ElasticSearch DBs. This defines the types and formats of the fields that appear in the documents. As a result, mapping can have a significant impact on how Elasticsearch searches for and stores data. After creating an index, we must define the mapping. An incorrect preliminary definition and mapping might lead to incorrect search results.

Types of mapping

- **Static mapping:** Users perform static mappings when they create an index. We use static mappings to define data types and indexes. It is easy to define fields and their types when creating an index.
- **Dynamic mapping:** Elasticsearch automatically creates dynamic mappings for the tables. The dynamic mapping of Elasticsearch comes in handy when we need to store extra attributes on documents. It is not always necessary to configure field names and types when indexing documents, as these will be created automatically by Elasticsearch based on any predefined rules.

7. What is ElasticSearch fuzzy search?

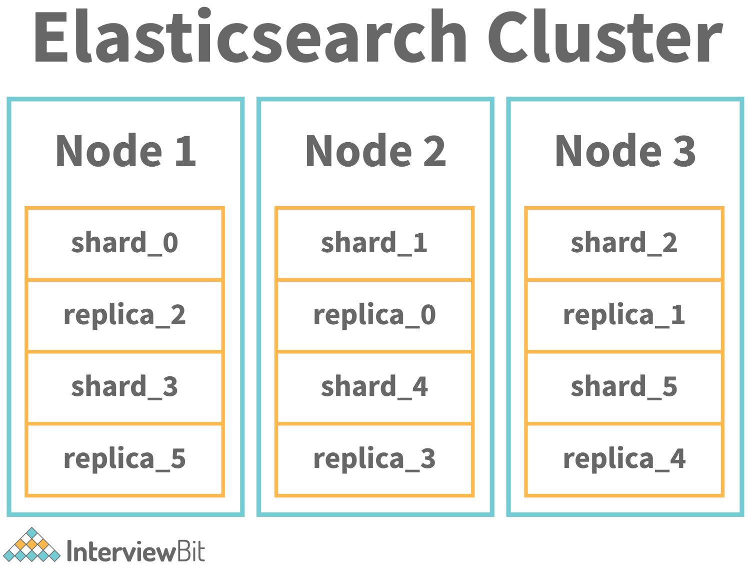
With fuzzy search, you can find documents with terms similar to your search term based on a Levenshtein edit distance measure. Edit distance is essentially the number of single-character changes or edits required to change one term into another. Among these changes are:

- Change one character (box → fox)
- Remove one character (black → lack)
- Insert one character (sic → sick)
- Transpose two adjacent characters (act → cat)

Within a specific edit distance, the fuzzy query generates a list of all possible variations and expansions of the search term. After that, the query returns a list of all possible matches. The most relevant and exact matches appear near the top of the list.

8. What is cluster in ElasticSearch?

A cluster is a collection of connected nodes. If you run only one instance or node of Elasticsearch, then you have a single-node cluster or a cluster of one node. Clusters automatically reorganize themselves when nodes join or leave so the data is distributed evenly among all the nodes. Despite being fully functional, the cluster is at risk of data loss if it fails.



9. Explain a node in Elastic Search.

You can think of a node as a single server that forms part of your cluster. Nodes are assigned roles that describe their responsibilities and operations. By default, every cluster node can handle HTTP and transport traffic. Communication between nodes is carried out via the transport layer, while REST clients utilize the HTTP layer. Nodes in a cluster are aware of each other and can forward client requests to the right node.

10. Explain what is a document in ElasticSearch.

The term "document" refers to a unit of information that can be indexed. Each index within Elasticsearch contains multiple documents. For instance, you could have a document for every customer, another for every order, etc. These documents are written in JavaScript Object Notation (JSON), which is a widely used format for internet data exchange. Documents are composed of fields, and each field has its own type of data. In a particular index, you can store as many documents as you wish.

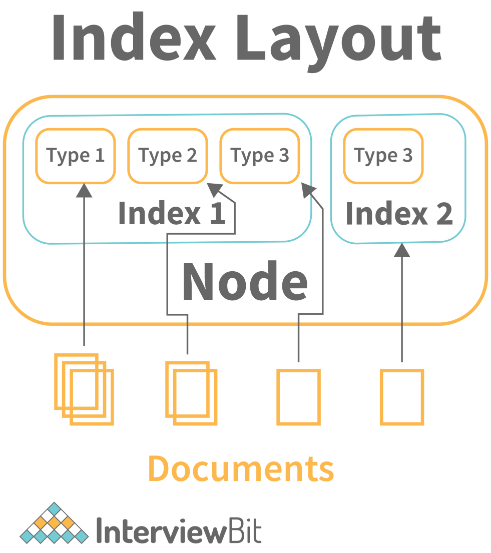
11. Which operations can you perform on a document?

Elasticsearch allows the following operations to be performed over documents:

- Indexing a document
- Fetching documents
- Updating documents
- Deleting documents

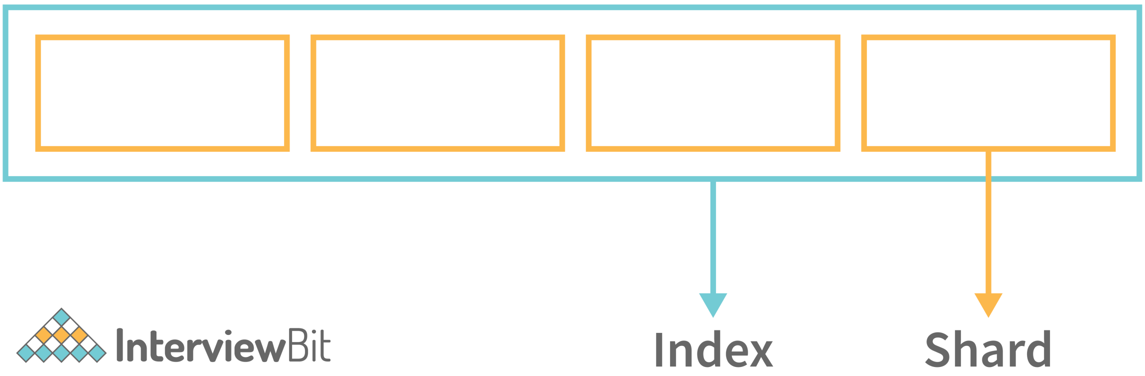
12. What is an index in ElasticSearch?

An index is a collection of documents that are somewhat similar in nature. As an example, you could have an index of customer data, another one of product catalogs, and another one of order data. The name of an index (which must be all lowercase) serves as an identifier for the index when indexing, searching, updating, and deleting documents contained within it. An index (plural: indices) can have one or more than one shards and replicas.



13. Define the terms Shard and Replica in ElasticSearch.

Shard: Elasticsearch crashes are often caused by large indexes. Due to the unlimited number of documents that can be stored on each index, an index may consume more disk space than the hosting server can provide. Indexing will begin to fail as soon as the index reaches this limit. As a solution to this problem, it is possible to divide or segment indexes horizontally into multiple pieces, also called shards. For an index, you can easily specify how many shards you want. As a result, every shard is its own, fully functional, and independent "index", which can run on any node in a cluster.



Replica: As the name implies, replicas are Elasticsearch fail-safe mechanisms, and are essentially copies of an index's shards. As a backup, this could come in handy when a node crashes. Furthermore, replicas can serve read requests, which is useful for increasing search performance. To ensure high availability, replicas must not be placed on the same node as the original shard (called the "primary shared") from which they were replicated.

14. What is the process of deleting an index in Elasticsearch?

Deleting an index removes all of its shards, documents, and metadata. Use the following command to delete an index in Elasticsearch:

Syntax:

```
DELETE /<index_name>
```

Example: To delete an index named my-index-321, we use the following command.

```
DELETE /my-index-321
```

15. How to add a mapping to an index?

Elasticsearch lets you add the mapping to an index based on the data provided by the user in the request body. The following syntax can be used to add a mapping to an index:

Syntax:

```
POST /_<index_name>/_type/_id
```

16. What do you mean by the term ‘type’ in ElasticSearch?

Types are logical categories or parts of an index whose semantics are determined by the user. Elasticsearch clusters can consist of multiple Indices (databases), each of which contains several Types (tables). A type holds multiple Documents (rows), and every document has some Properties (columns). By using types, multiple data types can be stored in the same index, thus reducing the total number of indices.

Example:

Suppose, in your car manufacturing scenario, you had a Tatafactory index. There are three types (tables) in this index as follows:

- Cars
- People
- Spare_Parts

Every type then contains documents relevant to that type (e.g. a Tata Innova document is housed in the Cars type). In this document, you can find all the information about the particular car.

17. What do you mean by NRT (Near Real-Time Search) in ElasticSearch?

Elasticsearch provides near real-time search functionality. It means that there is a slight delay (approximately one second) between the time you index a document and the moment it becomes searchable.

ElasticSearch Interview Questions for Experienced

18. Explain Tokenizer in ElasticSearch.

When a tokenizer receives a stream of characters (text), it tokenizes them (usually by breaking them up into individual words or tokens), and outputs the stream of words/tokens. Elasticsearch comes with several tokenizers that you can use to build your custom analyzers. A whitespace tokenizer, for example, breaks text into individual tokens whenever it encounters any whitespace. The text "Scaler by InterviewBit!" would be converted into terms or tokens [Scaler, by, InterviewBit].

19. What is an Analyzer ElasticSearch?

When indexing data in ElasticSearch, the data is internally transformed by the Analyzer assigned to the index. In essence, an analyzer indicates how text should be indexed and searched in ElasticSearch. Elasticsearch comes with several ready-to-use analyzers built into it. You can also create custom analyzers by combining the built-in character filters, tokenizers, and token filters.

- **Character filter:** Used to remove unused characters or change some characters.
- **Tokenizer:** Divides or breaks text into tokens (or words) based on some criteria (e.g. whitespace).
- **Token filter:** The filter receives tokens and applies filters to them (such as changing uppercase terms into lowercase).

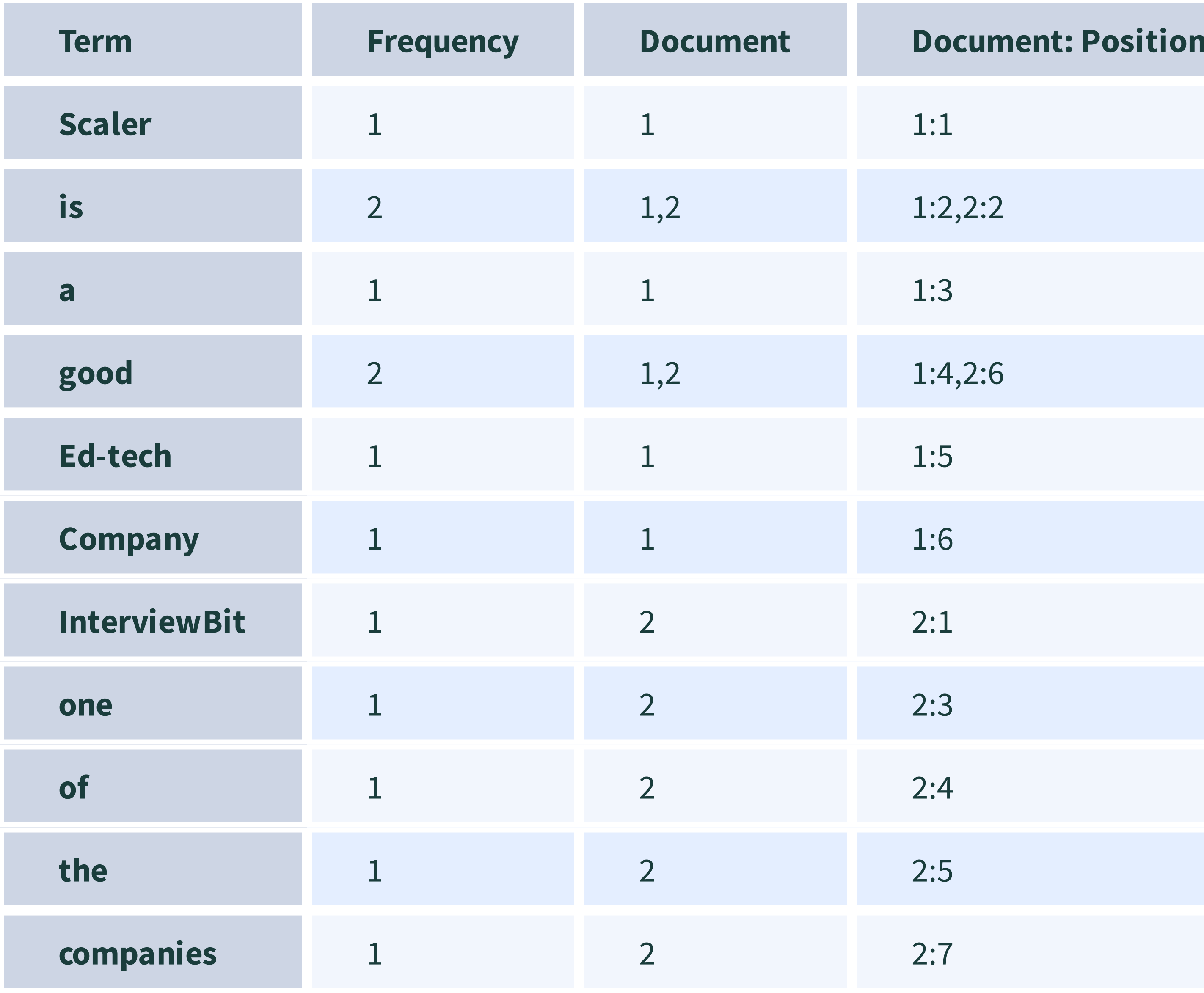
20. What is an Inverted index in ElasticSearch?

ElasticSearch utilizes a hashmap-like data structure known as an inverted index that allows for rapid full-text searches. The inverted index lists all the unique words that appear in one or more documents and identifies all the documents those words appear in. With it, you can conduct quick searches across millions of documents to find relevant data.

Example: Let's assume we have two different documents:

- Scaler is a good Ed-tech company.
- InterviewBit is one of the good companies.

The above texts have been tokenized first into separate terms for indexing purposes. All the unique terms are then stored in the index, along with information such as which document the term appears in, its position, as well as how many times it appeared. Accordingly, the inverted index is as follows:



Term	Frequency	Document	Document: Position
Scaler	1	1	1:1
is	2	1,2	1:2,2:2
a	1	1	1:3
good	2	1,2	1:4,2:6
Ed-tech	1	1	1:5
Company	1	1	1:6
InterviewBit	1	2	2:1
one	1	2	2:3
of	1	2	2:4
the	1	2	2:5
companies	1	2	2:7

Let's say you are looking for a term company or companies. With this inverted index, queries can search for terms and quickly identify documents that contain these terms.

21. Describe the functionality of the cat API in ElasticSearch.

Elasticsearch API results are usually displayed in JSON format, which is not always easy to read. Human eyes require compact and aligned text, especially when looking at a terminal. In order to meet this need, cat APIs (compact and aligned text APIs) have been developed. Thus, the cat APIs feature offered by Elasticsearch facilitates an easier-to-read and comprehend printing format for Elasticsearch results. Cat APIs return plain text instead of traditional JSON, which is comprehensible by users. You can view the available operations in the cat API by running the following commands:

```
GET _cat
```

Additionally, you may use the following parameters with your query.

- **Verbose (v?):** Gives results in a nice format or more verbose output. Use this parameter to see what each column represents.
  - **Syntax:**

```
GET _cat/<operation_name>?v
```

- **Help (?help):** Provides a list of the available headers and columns for a given operation. You can view all available headers by using this parameter.
  - **Syntax:**

```
GET _cat/<operation_name>?help
```

- **Headers (?h):** Limit the output to specified headers or columns in the command.
  - **Syntax:**

```
GET _cat/<operation_name>?h=<header_name_1>,<header_name_2>&v
```

- **Numeri format (?format):** Provide different types of numeric output, such as bytes, size, and time value.
- **Sort (?sort):** Sorts the table by the specified columns as the parameter value.

22. What are the different ElasticSearch commands available in the cat API?

There are different commands available in the Elasticsearch cat API. Here are a few:

- **Count:** Displays the total number of documents in your cluster.

```
GET _cat/count?v
```

- **Allocation:** Displays the disk space allocated to indices and the number of shards per node.

```
GET _cat/allocation?v
```

- **Field data**: Shows the memory usage of each field per node.

```
GET _cat/fielddata?v
```

- **Indices:** Displays information about indices, including how much space they take up, how many shards they have, etc.

```
GET _cat/indices?v
```

- **Node Attributes:** Displays the attributes associated with custom nodes.

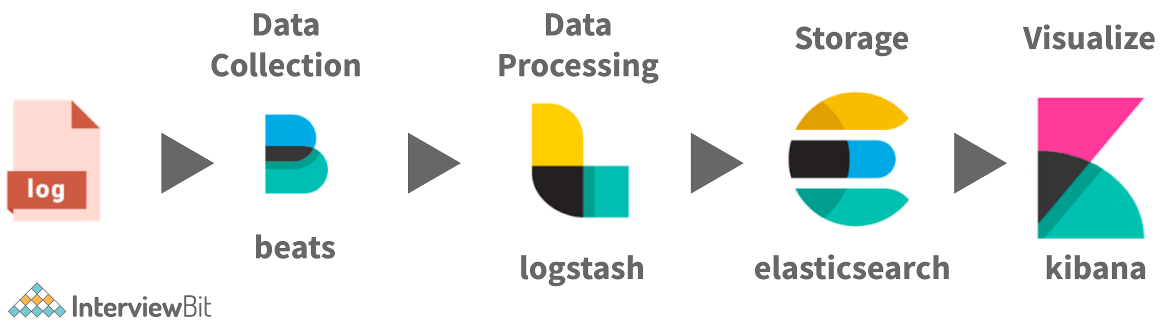
```
GET _cat/nodeattrs?v
```

23. Explain ELK stack and its architecture.

The "ELK" acronym refers to three open-source products i.e., Elasticsearch, Logstash, and Kibana, which are all produced, managed, and maintained by Elastic company. With the introduction of Beats, this stack became a four-legged project called BELK or Elastic Stack.

- **ElasticSearch**: Elasticsearch is an open-source, modern, full-text search engine based or built on Apache Lucene. ElasticSearch enables you to store, search, and analyze vast or huge amounts of data in near real-time, providing results in milliseconds.
- **Logstash:** Logstash is a data processing pipeline tool, which collects data from different sources, performs several transformations and enhancements, and then sends the data to stash or wherever you want it to go.
- **Kibana:** Kibana is a visualization tool built on top of Elasticsearch, enabling users to analyze and view data. The Kibana dashboard lets you visualize complex queries through interactive diagrams, geospatial data, and graphs.
- **Beats:** Beats are lightweight data shipping agents (data shippers) that collect different types of data and forward it to Logstash or ElasticSearch. There are different types of Beats, each focused on a specific type of data, such as metrics, log files, network packets, audit data, Windows events, uptime monitoring data, and cloud data.

These components are typically used in conjunction to monitor, troubleshoot, and secure IT environments. The Beats and Logstash tools handle the collection and processing of data, while Elasticsearch stores and indexes the data, and Kibana provides a graphical UI (user interface) for querying and visualizing the data.



24. What configuration management tools does Elasticsearch support?

Elasticsearch supports the following configuration management tools:

- **Chef:** cookbook-elastic search.
- **Puppet:** puppet-elastic search.
- **Ansible:** ansible-elastic search.

25. Is it necessary to install X-Pack for Elasticsearch? What are some essential X-pack commands?

Yes, if you are using ElasticSearch, you must install X-Pack. In essence, X-Pack is an Elastic Stack extension that combines or bundles alerting, reporting, monitoring, security, and graph capabilities into a single package that can be installed quickly and easily. Although the components of the X-Pack work seamlessly together, you can enable or disable the features you need. Since X-Pack is an Elastic Stack extension, you will need to install both Elasticsearch and Kibana before installing X-Pack. The version of X-Pack must match Elasticsearch and Kibana versions.

The following are a few X-Pack commands that can help you configure security and perform other tasks:

- elasticsearch-certgen
- elasticsearch-certutil
- elasticsearch-reset-password
- elasticsearch-setup-passwords
- elasticsearch-syskeygen
- elasticsearch-users, etc.

26. What do you mean by aggregation in ElasticSearch?

Aggregations in Elasticsearch enable you to group data and calculate statistics on your data with a simple search query. In ElasticSearch, aggregations are categorized into three types:

- **Bucket aggregations:** Documents can be grouped into buckets by using bucket aggregations. You can use them to create data buckets or group data. A bucket can be formed based on existing field values, ranges, etc.
- **Metric aggregations:** This aggregation helps to calculate metrics (such as a sum, or average) based on field values.
- **Pipeline aggregations:** This type of aggregation takes inputs from the output results of other aggregates rather than individual documents or fields.

27. Does ElasticSearch have a schema?

Yes, it is possible for ElasticSearch to have a schema. The schema is a description of one or more fields in a document that describe what type of document it is and how different fields of a document are to be handled. In Elasticsearch, a schema describes the fields in JSON documents, their data types, and how they should be indexed in the Lucene indexes. As such, we call this schema a "mapping" in Elasticsearch.ch.

But Elasticsearch can also be schema-less, meaning that documents can be indexed without the need to provide a schema explicitly. If a mapping isn't specified, Elasticsearch will by default generate one when newly added fields are detected during indexing.

28. How can we perform a search in Elasticsearch?

Below are a few ways to perform a search in Elasticsearch:

- **Applying search API:** You can search and aggregate data that is stored in Elasticsearch data streams and indices using the search API.
- **Search using a URI (Uniform Resource Identifier):** The search request is executed using a URI (Uniform Resource Identifier) by providing request parameters.
- **Request body search:** The search request should be executed using DSL (Domain Specific Language) within the body.

29. Explain Query DSL in ElasticSearch.

Elasticsearch generally provides a query Domain Specific Language (DSL) based on JSON to define queries. Query DSL contains two kinds of clauses:

- **Leaf query clause:** A leaf query clause looks for specific values in a field or fields. They can be used independently. Matches, terms, and range queries are some examples of these queries.
- **Compound query clause**: A compound query clause is a combination of a leaf query and other compound queries. These queries combine multiple queries to produce their intended results.

The behaviour of query clauses differs depending on whether it is used in a filter context or a query context.

30. What types of queries does ElasticSearch support?

Elasticsearch supports a wide range of queries. The query begins with a query keyword, which is followed by conditions and filters in the form of a JSON object. Here are a few of the queries:

- **Match All Query:** This is a basic query that retrieves all the documents in the specified index.
- **Full-text queries:** There are high-level queries for executing full-text searches over full-text fields. Full-text queries usually work depending on the analyzer associated with a particular document or index. Full-text queries can be of different types, such as match query, multi-match query, query-string query, etc.
- **Term Level Queries:** Instead of full-text field searches, term-level queries deal with structured data like numbers, enums, dates, etc. Term level queries can be of different types, such as range, exists, prefix, wildcard, fuzzy, type, etc.

Conclusion

In recent years, it has evolved into one of the most popular search engines that are used for business analytics, log analytics, security intelligence, operational intelligence, full-text searches, etc. Research has shown that Elasticsearch has a market share of about 0.24%, so there is a lot of opportunity for many renowned companies. Thus, you are still able to advance in your career as an ElasticSearch Engineer. Almost every area of ElasticSearch, along with the ELK stack, has been covered in the interview questions, including questions about the analyzers, filters, tokenizers, index, token filters, and APIs used in ElasticSearch.

Here, we have compiled a list of insightful interview questions that give ample information vital to the interview process. Being familiar with these frequently asked interview questions increases your chances of getting hired.

Hopefully, we have answered any questions or concerns that you may have had. All the best with your future endeavours.

Additional Interview Preparation Resources

- AWS Interview Questions
- SQL Interview Questions
- REST API Interview Questions
- Other Technical Interview Questions

Links to More Interview Questions

- C Interview Questions
- Web Api Interview Questions
- Cpp Interview Questions
- Machine Learning Interview Questions
- Css Interview Questions
- Django Interview Questions
- Operating System Interview Questions
- Git Interview Questions
- Dbms Interview Questions
- Pl Sql Interview Questions
- Ansible Interview Questions
- Php Interview Questions
- Hibernate Interview Questions
- Oops Interview Questions
- Docker Interview Questions
- Laravel Interview Questions
- Dot Net Interview Questions
- React Native Interview Questions
- Java 8 Interview Questions
- Spring Boot Interview Questions
- Tableau Interview Questions
- Java Interview Questions
- C Sharp Interview Questions
- Node Js Interview Questions
- Devops Interview Questions
- Mysql Interview Questions
- Asp Net Interview Questions
- Kubernetes Interview Questions
- Aws Interview Questions
- Mongodb Interview Questions
- Power Bi Interview Questions
- Linux Interview Questions
- Jenkins Interview Questions