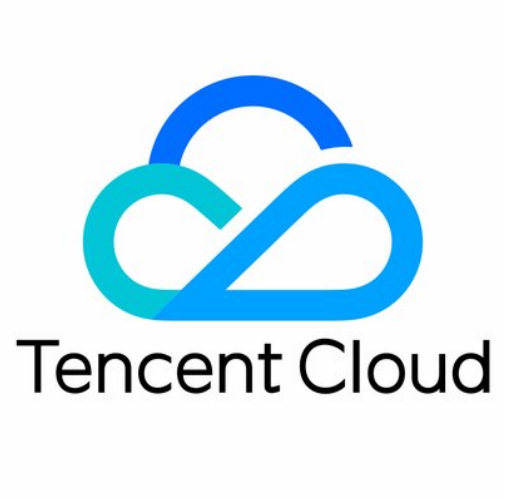
Elasticsearch Service

Glossary

Product Documentation



Glossary

Last updated：2019-11-08 17:42:27

Shard

Shards are data containers where documents are stored. A shard is an underlying work unit, which only stores part of the data. Shards are allocated to different nodes in a cluster. When you scale your cluster, Elasticsearch will automatically migrate shards among nodes to distribute the data evenly in the cluster.
A shard can be either a primary shard or a replica shard. Any document in an index belongs to a primary shard, so the number of primary shards determines the maximum volume of data that can be stored in the index. Technically, a primary shard can hold up to 128 documents (Integer.MAX_VALUE).
A replica shard is just a copy of a primary shard. It acts as a redundant backup that protects data from loss when hardware fails and serves read operations such as searching for and returning documents. The number of primary shards is configured during index creation, but the number of replica shards can be modified at any time.

Cluster and node

A running Elasticsearch instance is called a node. One or more nodes in the same network that have the same cluster name and have network communication configured form an Elasticsearch cluster. The nodes in the cluster sustain data storage and query requests together. When a node is added to or removed from the cluster, the cluster will re-distribute all the data evenly. Each node knows where any document is located. Therefore, regardless of which node the user sends requests to, the requests can be forwarded directly to the nodes where the required documents are stored, and the data can be collected from each node storing the required documents to eventually return the result to the client.

Index

Common concepts

An index is similar to a data table in a traditional relational database and is where relational documents are stored.
To index a document is to store a document to an index, so that it can be searched for and queried. Except the circumstance where existing documents will be overwritten by new documents with the same name, this operation is very like the INSERT keyword in SQL statements.

Inverted index

A relational database speeds up data search by adding an index (e.g., B-tree index) to a specified column. Elasticsearch and Lucene use an inverted index structure to achieve the same goal. By default, every attribute of a document is indexed (with an inverted index) and searchable. Attributes with no inverted indices cannot be searched.

Document

Elasticsearch is document-oriented. It stores entire objects or documents and indexes the contents of each document to make them searchable. It uses JSON as the serialization format of documents, making them simple, concise, and easy to read. JSON serialization is supported by most programming languages and has become a standard format in the NoSQL world. In Elasticsearch, instead of searching for data based on row and column, you index, search for, sort, and filter documents. This is a completely different way of thinking about data and exactly why Elasticsearch can support complex full-text searches.