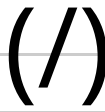




---

[Docs \(/guide\)](#)

---

[Docs \(/guide\)](#)

You are looking at documentation for an older release. Not what you want? See the current release documentation ([../current/index.html](#)).

[Elasticsearch Reference \[5.3\] \(index.html\)](#) » [Getting Started \(getting-started.html\)](#) » Basic Concepts

« [Getting Started \(getting-started.html\)](#)

[Installation » \(\\_installation.html\)](#)

## Basic Concepts

[edit \(https://github.com/elastic/elasticsearch/edit/5.3/docs/reference/getting-started.asciidoc\)](https://github.com/elastic/elasticsearch/edit/5.3/docs/reference/getting-started.asciidoc)

There are a few concepts that are core to Elasticsearch. Understanding these concepts from the outset will tremendously help ease the learning process.

### Near Realtime (NRT)

[edit \(https://github.com/elastic/elasticsearch/edit/5.3/docs/reference/getting-started.asciidoc\)](https://github.com/elastic/elasticsearch/edit/5.3/docs/reference/getting-started.asciidoc)

Elasticsearch is a near real time search platform. What this means is there is a slight latency (normally one second) from the time you index a document until the time it becomes searchable.

### Cluster

[edit \(https://github.com/elastic/elasticsearch/edit/5.3/docs/reference/getting-started.asciidoc\)](https://github.com/elastic/elasticsearch/edit/5.3/docs/reference/getting-started.asciidoc)

A 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.

Make sure that you don't reuse the same cluster names in different environments, otherwise you might end up with nodes joining the wrong cluster. For instance you could use `logging-dev`, `logging-stage`, and `logging-prod` for the development, staging, and production clusters.

Note that it is valid and perfectly fine to have a cluster with only a single node in it. Furthermore, you may also have multiple independent clusters each with its own unique cluster name.

## Node

[edit \(https://github.com/elastic/elasticsearch/edit/5.3/docs/reference/getting-started.asciidoc\)](https://github.com/elastic/elasticsearch/edit/5.3/docs/reference/getting-started.asciidoc)

A node is a single server that is part of your cluster, stores your data, and participates in the cluster's indexing and search capabilities. Just like a cluster, a node is identified by a name which by default is a random Universally Unique Identifier (UUID) that is assigned to the node at startup. You can define any node name you want if you do not want the default. This name is important for administration purposes where you want to identify which servers in your network correspond to which nodes in your Elasticsearch cluster.

A node can be configured to join a specific cluster by the cluster name. By default, each node is set up to join a cluster named `elasticsearch` which means that if you start up a number of nodes on your network and—assuming they can discover each other—they will all automatically form and join a single cluster named `elasticsearch`.

In a single cluster, you can have as many nodes as you want. Furthermore, if there are no other Elasticsearch nodes currently running on your network, starting a single node will by default form a new single-node cluster named `elasticsearch`.

## Index

[edit \(https://github.com/elastic/elasticsearch/edit/5.3/docs/reference/getting-started.asciidoc\)](https://github.com/elastic/elasticsearch/edit/5.3/docs/reference/getting-started.asciidoc)

An index is a collection of documents that have somewhat similar characteristics. For example, you can have an index for customer data, another index for a product catalog, and yet another index for order data. An index is identified by a name (that must be all lowercase) and this name is used to refer to the index when performing indexing, search, update, and delete operations against the documents in it.

In a single cluster, you can define as many indexes as you want.

## Type

[edit \(https://github.com/elastic/elasticsearch/edit/5.3/docs/reference/getting-started.asciidoc\)](https://github.com/elastic/elasticsearch/edit/5.3/docs/reference/getting-started.asciidoc)

Within an index, you can define one or more types. A type is a logical category/partition of your index whose semantics is completely up to you. In general, a type is defined for documents that have a set of common fields. For example, let's assume you run a blogging platform and store all your data in a single index. In this index, you may define a type for user data, another type for blog data, and yet another type for comments data.

## Document

[edit \(https://github.com/elastic/elasticsearch/edit/5.3/docs/reference/getting-started.asciidoc\)](https://github.com/elastic/elasticsearch/edit/5.3/docs/reference/getting-started.asciidoc)

A document is a basic unit of information that can be indexed. For example, you can have a document for a single customer, another document for a single product, and yet another for a single order. This document is expressed in JSON (<http://json.org/>) (JavaScript Object Notation) which is an ubiquitous internet data interchange format.

Within an index/type, you can store as many documents as you want. Note that although a document physically resides in an index, a document actually must be indexed/assigned to a type inside an index.

## Shards & Replicas

[edit \(https://github.com/elastic/elasticsearch/edit/5.3/docs/reference/getting-started.asciidoc\)](https://github.com/elastic/elasticsearch/edit/5.3/docs/reference/getting-started.asciidoc)

An index can potentially store a large amount of data that can exceed the hardware limits of a single node. For example, a single index of a billion documents taking up 1TB of disk space may not fit on the disk of a single node or may be too slow to serve search requests from a single node alone.

To solve this problem, Elasticsearch provides the ability to subdivide your index into multiple pieces called shards. When you create an index, you can simply define the number of shards that you want. Each shard is in itself a fully-functional and independent "index" that can be hosted on any node in the cluster.

Sharding is important for two primary reasons:

- It allows you to horizontally split/scale your content volume
- It allows you to distribute and parallelize operations across shards (potentially on multiple nodes) thus increasing performance/throughput

The mechanics of how a shard is distributed and also how its documents are aggregated back into search requests are completely managed by Elasticsearch and is transparent to you as the user.

In a network/cloud environment where failures can be expected anytime, it is very useful and highly recommended to have a failover mechanism in case a shard/node somehow goes offline or disappears for whatever reason. To this end, Elasticsearch allows you to make one or more copies of your index's shards into what are called replica shards, or replicas for short.

Replication is important for two primary reasons:

- It provides high availability in case a shard/node fails. For this reason, it is important to note that a replica shard is never allocated on the same node as the original/primary shard that it was copied from.
- It allows you to scale out your search volume/throughput since searches can be executed on all replicas in parallel.

To summarize, each index can be split into multiple shards. An index can also be replicated zero (meaning no replicas) or more times. Once replicated, each index will have primary shards (the original shards that were replicated from) and replica shards (the copies of the primary shards). The number of shards and replicas can be defined per index at the time the index is created. After the index is created, you may change the number of replicas dynamically anytime but you cannot change the number of shards after-the-fact.

By default, each index in Elasticsearch is allocated 5 primary shards and 1 replica which means that if you have at least two nodes in your cluster, your index will have 5 primary shards and another 5 replica shards (1 complete replica) for a total of 10 shards per index.

**NOTE**

Each Elasticsearch shard is a Lucene index. There is a maximum number of documents you can have in a single Lucene index. As of `LUCENE-5843` (<https://issues.apache.org/jira/browse/LUCENE-5843>), the limit is 2,147,483,519 (= `Integer.MAX_VALUE - 128`) documents. You can monitor shard sizes using the `_cat/shards` (`cat-shards.html`) api.

With that out of the way, let's get started with the fun part...

---

« [Getting Started \(getting-started.html\)](#)

[Installation » \(\\_installation.html\)](#)

## Top Videos

- [Elasticsearch Demo \(https://www.elastic.co/webinars/getting-started-elasticsearch?baymax=default&elektra=docs&storm=top-video\)](https://www.elastic.co/webinars/getting-started-elasticsearch?baymax=default&elektra=docs&storm=top-video)
- [Kibana 101 \(https://www.elastic.co/webinars/getting-started-kibana?baymax=default&elektra=docs&storm=top-video\)](https://www.elastic.co/webinars/getting-started-kibana?baymax=default&elektra=docs&storm=top-video)
- [Logstash Primer \(https://www.elastic.co/webinars/getting-started-logstash?baymax=default&elektra=docs&storm=top-video\)](https://www.elastic.co/webinars/getting-started-logstash?baymax=default&elektra=docs&storm=top-video)

---

## On this page

[Near Realtime \(NRT\)](#)

[Cluster](#)

[Node](#)

[Index](#)

[Type](#)

[Document](#)

[Shards & Replicas](#)

---

Elasticsearch Reference: 5.3 ▼

[Getting Started \(getting-started.html\)](#)

**[Basic Concepts \(\\_basic\\_concepts.html\)](#)**

Installation ( <a href="#">_installation.html</a> )
Exploring Your Cluster ( <a href="#">_exploring_your_cluster.html</a> )
Modifying Your Data ( <a href="#">_modifying_your_data.html</a> )
Exploring Your Data ( <a href="#">_exploring_your_data.html</a> )
Conclusion ( <a href="#">_conclusion.html</a> )
Setup Elasticsearch ( <a href="#">setup.html</a> )
Breaking changes ( <a href="#">breaking-changes.html</a> )
API Conventions ( <a href="#">api-conventions.html</a> )
Document APIs ( <a href="#">docs.html</a> )
Search APIs ( <a href="#">search.html</a> )
Aggregations ( <a href="#">search-aggregations.html</a> )
Indices APIs ( <a href="#">indices.html</a> )
cat APIs ( <a href="#">cat.html</a> )
Cluster APIs ( <a href="#">cluster.html</a> )
Query DSL ( <a href="#">query-dsl.html</a> )
Mapping ( <a href="#">mapping.html</a> )
Analysis ( <a href="#">analysis.html</a> )
Modules ( <a href="#">modules.html</a> )
Index Modules ( <a href="#">index-modules.html</a> )
Ingest Node ( <a href="#">ingest.html</a> )
How To ( <a href="#">how-to.html</a> )
Testing ( <a href="#">testing.html</a> )
Glossary of terms ( <a href="#">glossary.html</a> )
Release Notes ( <a href="#">es-release-notes.html</a> )
A. Painless API Reference ( <a href="#">painless-api-reference.html</a> )

Be in the know with the latest and greatest from Elastic.