CHAPTER-1

Inverted indexing

Relevancy score (ranking)

TF(Term frequency)

IDF(Inverse document frequency)

Typo

JSON and YAML

REST API

CHAPTER-2

Logical layout

Documents

Self-Contained

Hierarchical

Flexible structure (Schema-Free)

Types

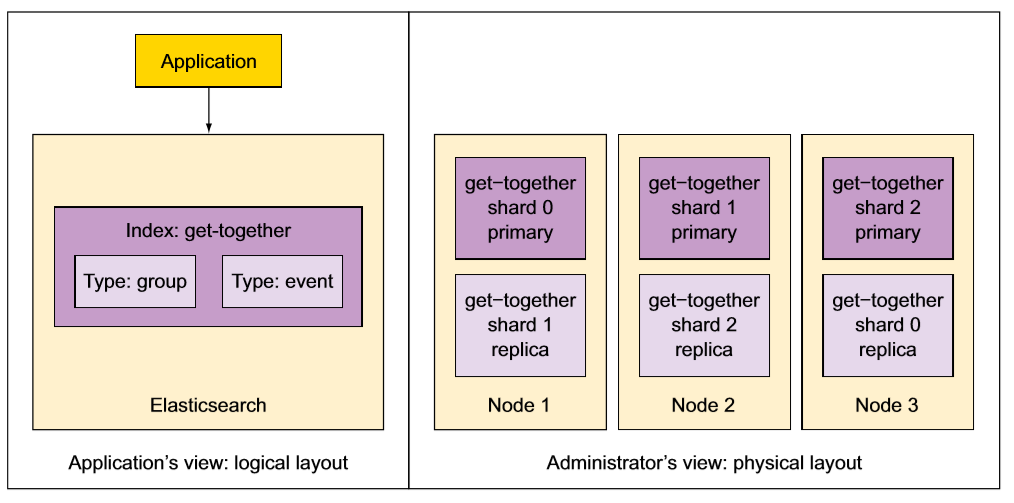
Mapping

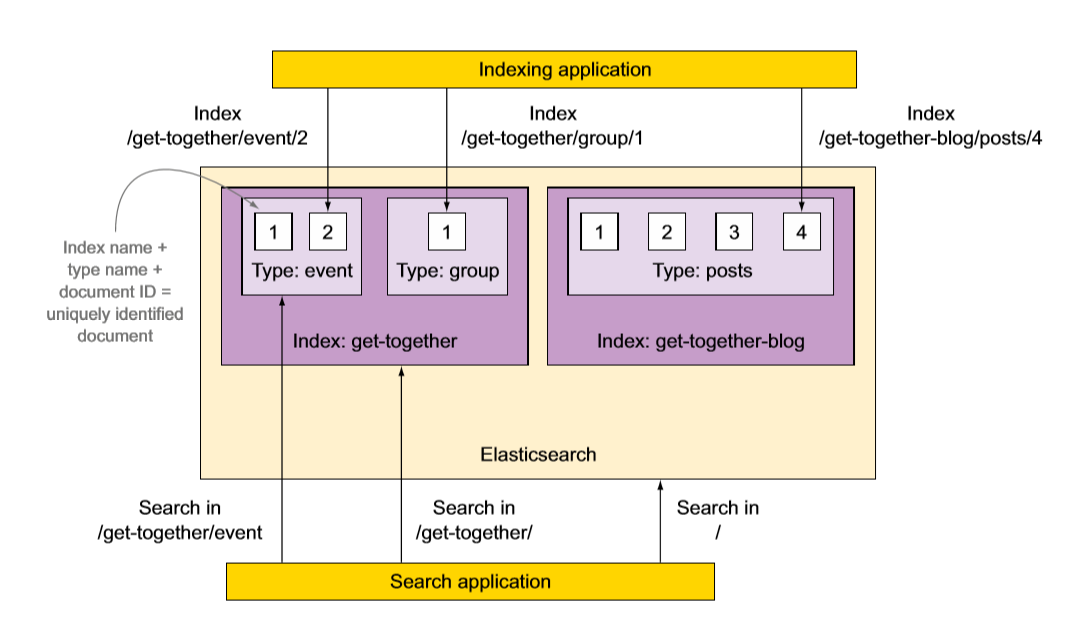
Indices

Shard (indices divided into parts)

refresh\_interval (near real time)

Elasticsearch





Physical Layout

Shards

Elasticsearch index is broken down into chunks called shards.

Shard is Lucene index: directory of files containing an inverted index.

Elasticsearch index is made up of multiple Lucene indices.

Inverted index is a structure that enables Elasticsearch to tell you which document contains a term without having to look at all documents.

Shard is smallest unit that Elasticsearch moves from node and node

Default 🡪 index = 5 primary shard + its replica (10)

Node

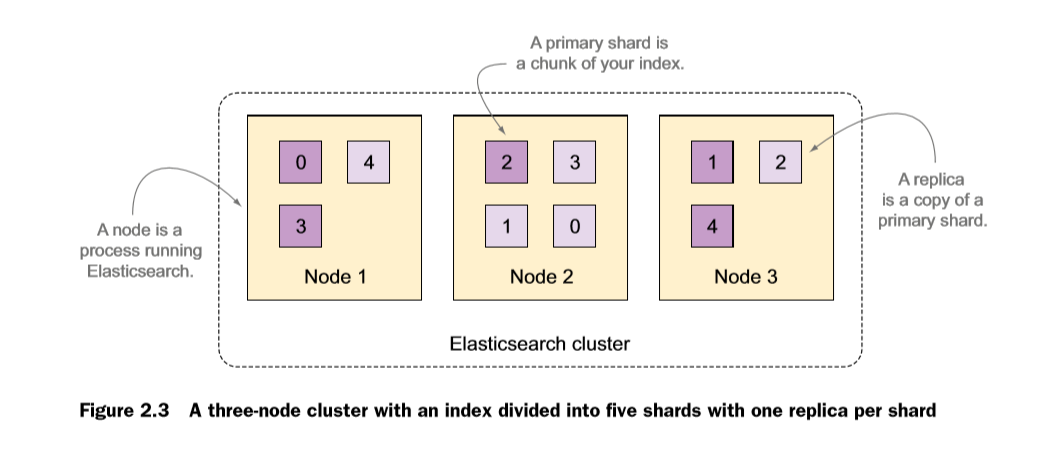
Node is instance of Elasticsearch

Start Elasticsearch on your server, you have node

Different server, different node

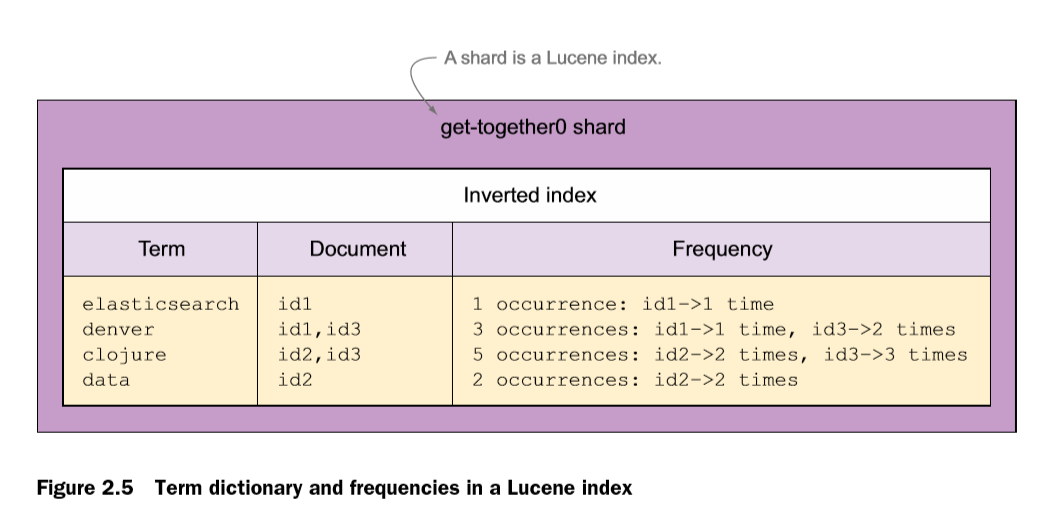
More nodes on same server, by starting multiple Elasticsearch processes

Multiple nodes = cluster



Shard = original document content + term dictionary

+ term frequencies



An Elasticearch index is made up of one or more primary shards and zero or more replica shard.

All this shards and its replica are distributed to nodes within Elasticsearch cluster.

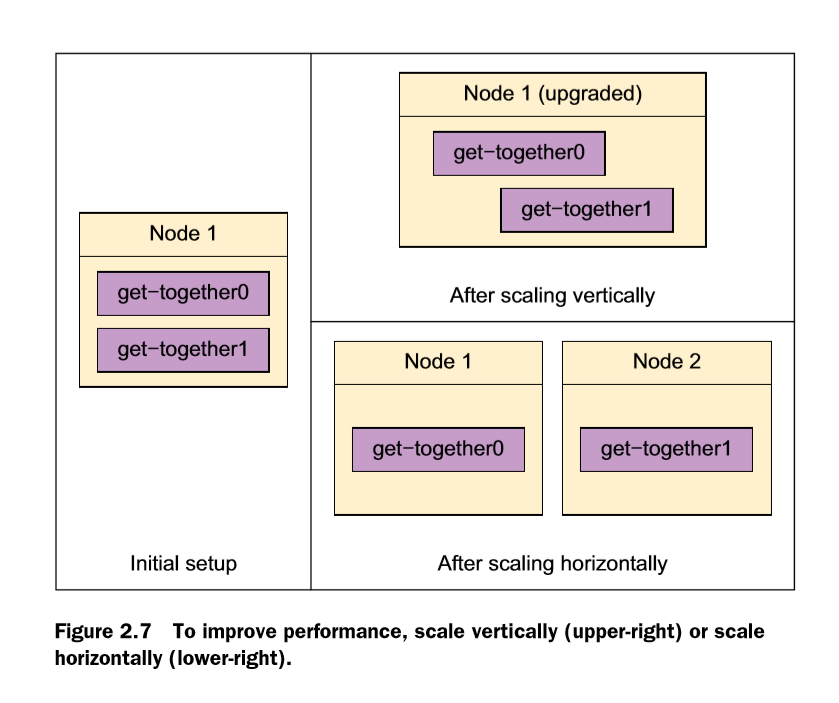
Simplest Elasticsearch cluster has one node.

You add more nodes to same cluster, existing shards get balances between all nodes.

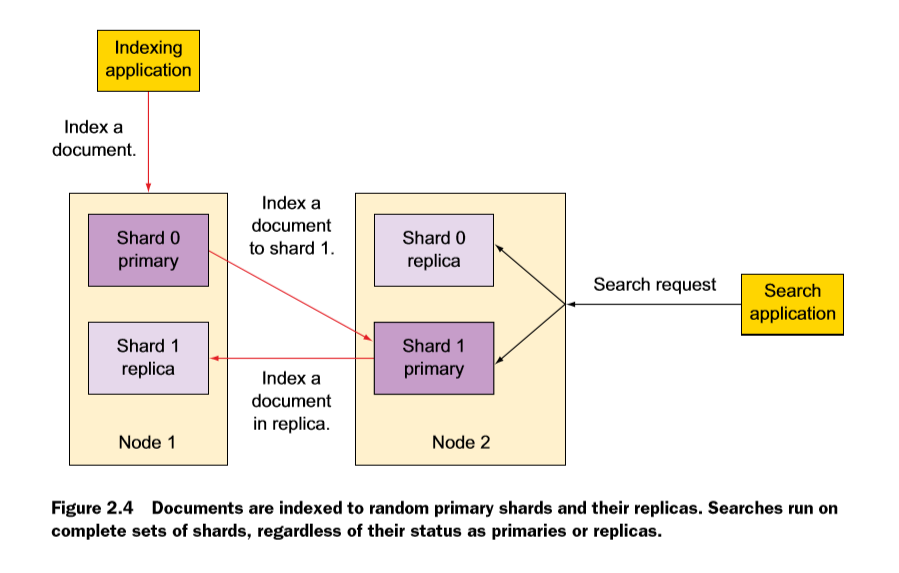
More nodes = strong searching and indexing

Adding nodes to cluster is called horizontal scaling

Vertical scaling = add more resources to your node , by dedicating more processors to it, or adding RAM to physical machine. (Cost effective)



Indexing and Searching



Indexing:

Request 🡪 Elasticsearch node 🡪 Select shard

Shard is determined by hashing its ID

Current node 🡪 node (Selected shard) and send to all replicas of that shard

Searching:

Searching 🡪 node 🡪 shard

Using round-robin 🡪 select shard(p+r) and forward search request

Gathers results from shards, aggregate them into single reply , forward it to application

Types of Queries:

(1)match : from, to, operator(and, or),

minimum-should-match(percentage)

(2)multi-match

(3)Boosting (revalency score)

(4)Bool (should, must, must\_not)

term (exact that word, case-sensetive)

(5)Fuzziness

(6)wildcard

(7)regular expression (regexp)

(8)match\_phrase :- slop

(9)Query string

(10)simple query string

(11)term or terms :- sort

(12)range :- lte ,gte

(13)filtered bool , filter

(14)function score , function value fator, decay functions, script scoring

(15)function score , decay functions