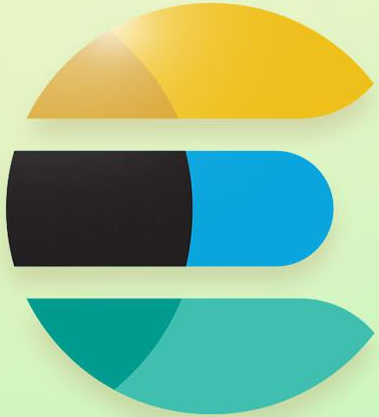
# Introduction to full text queries



codingexplained.com

youtube.com/@codingexplained

CodingExplained

CODE. SLEEP. REPEAT.

# Introduction

- Term level queries are used for exact matching on structured data
- Full text queries are used for searching unstructured text data
  - E.g. website content, news articles, emails, chats, transcripts, etc.
  - Often used for long texts
  - We don’t know which values a may field contain (hence “unstructured”)

# Understanding Sharding in Elasticsearch

title

Published on August 8, 2017 by Bo Andersen

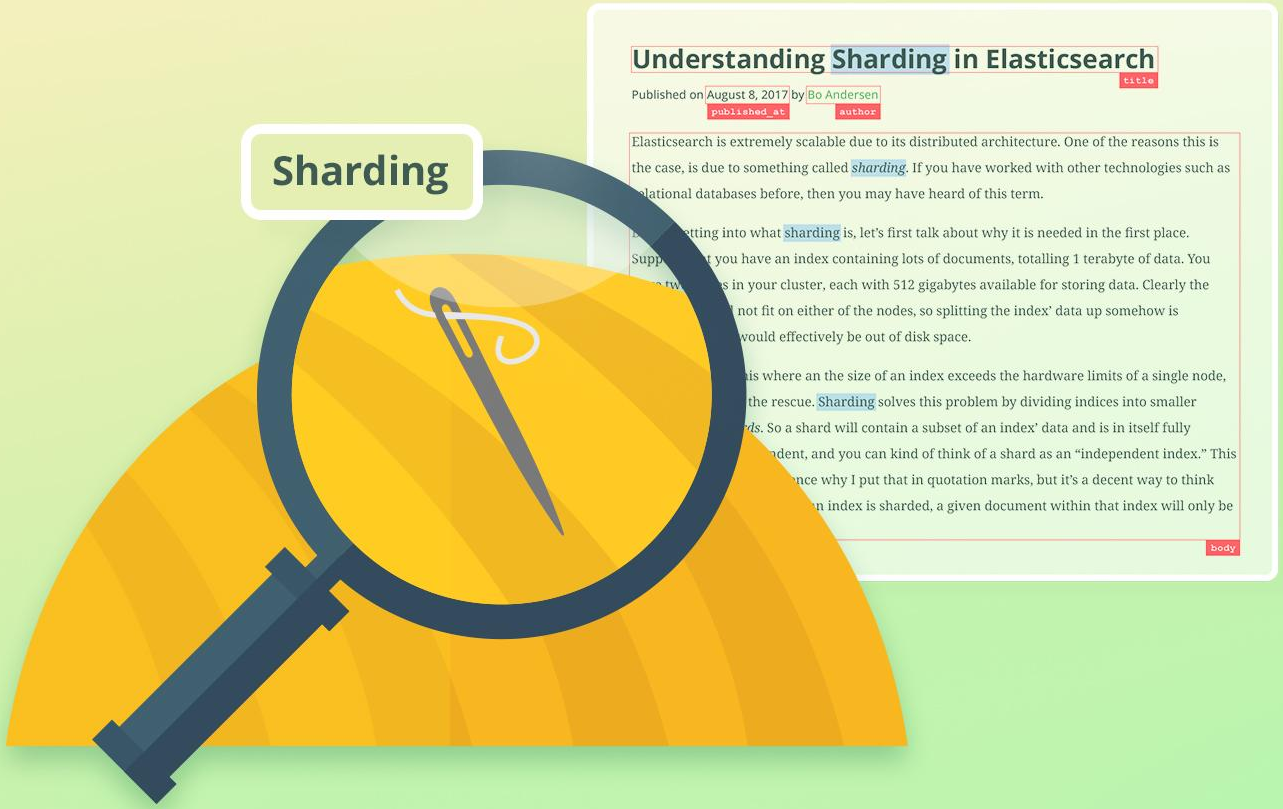
published_at author

Elasticsearch is extremely scalable due to its distributed architecture. One of the reasons this is the case, is due to something called *sharding*. If you have worked with other technologies such as relational databases before, then you may have heard of this term.

Before getting into what sharding is, let's first talk about why it is needed in the first place. Suppose that you have an index containing lots of documents, totalling 1 terabyte of data. You have two nodes in your cluster, each with 512 gigabytes available for storing data. Clearly the entire index will not fit on either of the nodes, so splitting the index' data up somehow is necessary, or we would effectively be out of disk space.

In scenarios like this where an the size of an index exceeds the hardware limits of a single node, sharding comes to the rescue. Sharding solves this problem by dividing indices into smaller pieces named *shards*. So a shard will contain a subset of an index' data and is in itself fully functional and independent, and you can kind of think of a shard as an "independent index." This is not entirely accurate, hence why I put that in quotation marks, but it's a decent way to think about it nevertheless. When an index is sharded, a given document within that index will only be stored within one of the shards.

body




CodingExplained
CODE. SLEEP. REPEAT.

# Full text queries are analyzed

## Query

```
GET /products/_search
{
  "query": {
    "match": {
      "body": "SHARDING"
    }
  }
}
```

"SHARDING"


"sharding"

## Inverted index

| TERM | DOCUMENT #1 |
|---|---|
| "elasticsearch" | X |
| "distributed" | X |
| "sharding" | X |
| "replication" | X |
| "nodes" | X |

# Full text queries vs term level queries

- The main difference is that full text queries are analyzed
  - Term level queries aren't and are therefore used for exact matching
- Don't use full text queries on `keyword` fields
  - That compares analyzed values with non-analyzed values

# Lecture summary

- Use full text queries to search unstructured text values
  - E.g. blog posts, emails, chats, books, song titles, etc.
- Full text queries are not used for exact matching
  - They match values that *include* a term, often being one of many
- Full text queries are analyzed in the same way as the fields that are queried
  - Don't query `keyword` fields with full text queries because the field values were not analyzed during indexing