# Searching Your Data with Elasticsearch

MICHAEL HOFFMAN

github.com/michaelhoffmantech

# Elasticsearch Demo Project

**github.com/michaelhoffmantech/elasticsearch-demo**

Search is one of the primary ways we interact with applications to find relevant data

Search results often fail at delivering relevant results

Search is hard!

We don’t always use the right tools

Product requirements for search are often incomplete or incorrect

Search engines were very complex

Search engines evolving and improving with more features

We are getting better at delivering the right search results

1-24 of 111 results for **Toys & Games** : **"exploding kittens"**

Sort by Relevance

**FREE Shipping**

All customers get FREE Shipping on orders over $25 shipped by Amazon

## Show results for

Any Category
- **Toys & Games**
  - Card Games
  - Grown-Up Toys
  - Board Games
  - Games
  - Action & Toy Figures
  - Cases
  - Stuffed Animals & Plush Toys
  - See more

## Refine by

**Delivery Day**
- Get It Today
- Get It by Tomorrow

**Amazon Prime**
- prime

**Eligible for Free Shipping**
- Free Shipping by Amazon

**Brand**
- Exploding Kittens LLC
- U.C.C. Distributing
- Breaking Games
- Gamewright



Showing results in **Toys & Games**. Show instead results in All Departments.

Summer Toy List


See more choices

Sponsored

You've Got Crabs: A Card Game From the Creators of Exploding Kittens

$24.99 prime

Get it by **Saturday, Jul 21**

FREE Shipping on eligible orders

Manufacturer recommended age: 7 - 85 Years

Show only Exploding Kittens LLC items

Amazon's Choice


See more choices

Exploding Kittens Card Game

$19.99 prime

Get it by **TODAY, Jul 19**

FREE Shipping on eligible orders

Manufacturer recommended age: 7 Years and up

Show only Exploding Kittens LLC items

7,433


See more choices

Exploding Kittens: NSFW Edition (Explicit Content - ADULTS ONLY!)

$19.99 prime

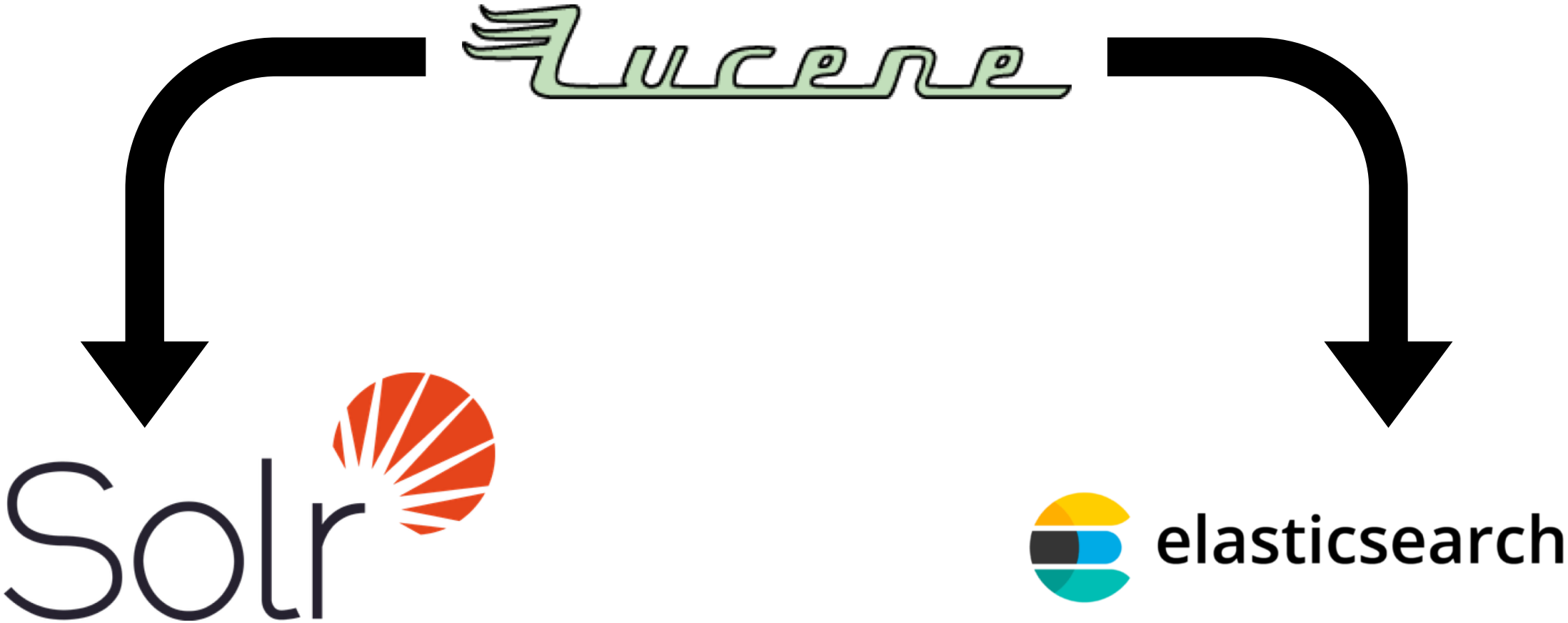
Get it by **TODAY, Jul 19**

FREE Shipping on eligible orders

Manufacturer recommended age: 16 Years and up

Show only Exploding Kittens LLC items

4,189


Lucene
Solr
elasticsearch

# Elasticsearch Features


- Search and analytics engine
- API-driven
- Document-based
- Analyzers
- Search query DSL
- Aggregations

# Demo

- Start the Elasticsearch Docker image
- Interact with Kibana

# Elasticsearch Runtime


- AWS Elasticsearch Service
- Elasticsearch Service
- Standalone
- Container

# Elasticsearch - Shard


- Lucene index
- Hosted on an Elasticsearch Node
- Horizontal scaling of data
- Replicas provide failover, H/A

# Elasticsearch - Node


- Single Elasticsearch server
- Part of an Elasticsearch cluster
- Manages cluster communication

# Elasticsearch - Cluster


- Collection of nodes
- Unique name where nodes register

# Docker - elasticsearch.yml

```
services:
  foodfacts-elasticsearch:
    image: docker.elastic.co/elasticsearch/elasticsearch:6.4.0
    environment:
      - cluster.name=docker-cluster
    ports:
      - 9200:9200
      - 9300:9300
  foodfacts-elasticsearch2:
  foodfacts-kibana:
    image: docker.elastic.co/kibana/kibana:6.4.0
    port:
      - 5601:5601
```

# Use capacity planning to determine scale

# Sample Case – Scale

- AWS Elasticsearch 5.3
- 4 x m4.xlarge
- 80M+ documents

# Demo

- Create an index
- Load data into the index

# Data Source

https://static.openfoodfacts.org/data/data-fields.txt

# Elasticsearch - Index


- Collection of similar documents
- Settings
- Mappings
- Fields

# food_product_index_v1

```
{
  “settings”: {
    “number_of_shards”: 2,
    “number_of_replicas”: 1
  },
  “mappings”: {
    “product”: {
      “properties”: {
      }
    }
  }
}
```

```
"code": {
    "type": "keyword",
    "index": "false"
}
```

food_product_index_v1.json

```
"productName": {
    "type": "text"
}
```

# food_product_v1.json

Found in resources folder of the project

```
"brands": {
      "type": "nested",
      "properties": {
            "name": {
                  "type": "text"
            }
      }
}
```

# food_product_v1.json

Found in resources folder of the project

# Elasticsearch - Document


- Contains the data
- Conforms to an index mapping
- Structured as JSON

# Elasticsearch - Types


- Logical partitioning of an index
- Required and restricted
- Deprecated

# Sample Case – Structure

- Used for NoSQL DB and search
- 12 indexes for domain entities
- 2 indexes with aggregated data

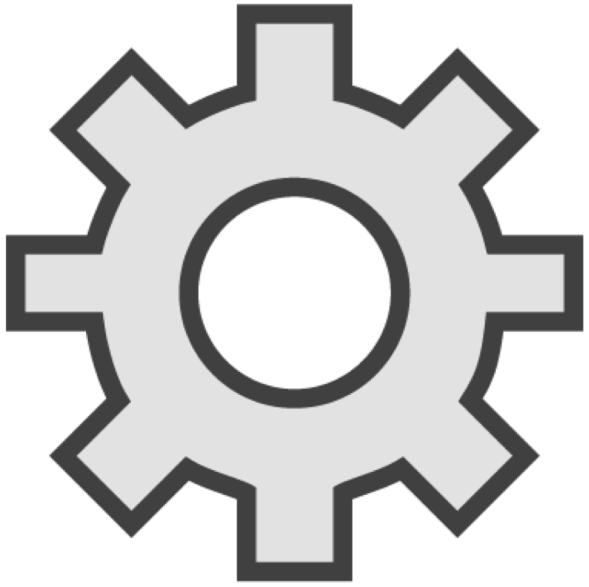
# Process for Indexing Content

# Extract data from a source

```
70847811169,Monster Energy,monster monster-energy
```

openfoodfacts_search.csv

# Data Extraction Considerations



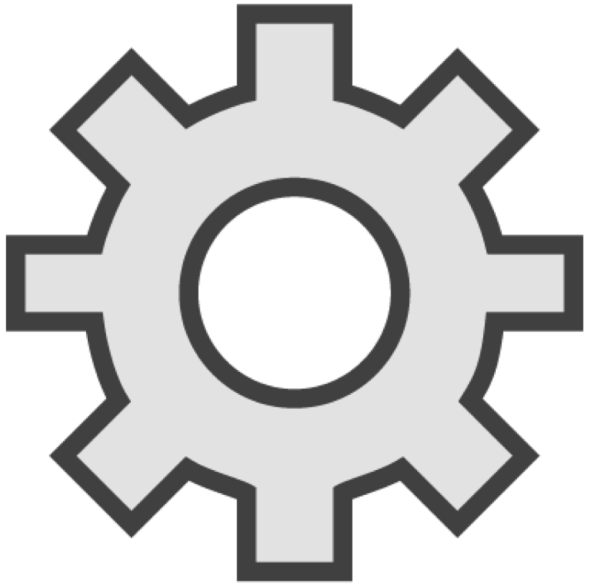
- Streaming input
- Batch input
- Source of truth

# Enrich the data

```
{
  "code":"00708478111169”,
  "productName":"Monster Energy",
  "brands":[
    {"name":"monster"},{"name":"monster-energy"}
  ]
}
```


openfoodfacts.txt

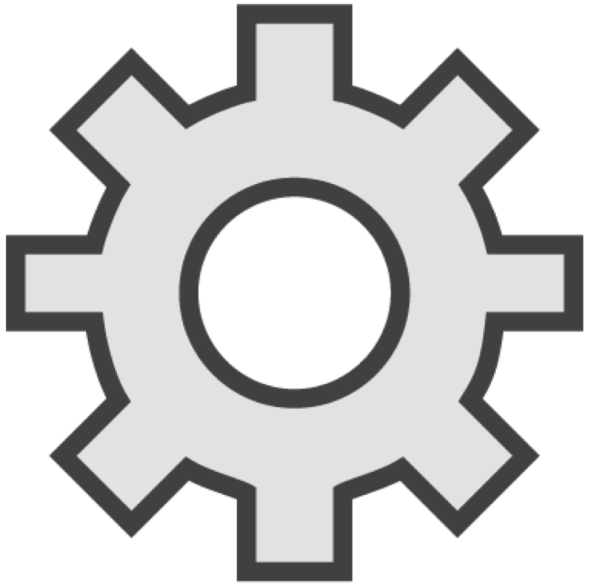
# Data Enrichment Considerations



- Index only what you need
- Derived fields
- Use analyzers when possible

# Analyze the data

# Analysis Breakdown



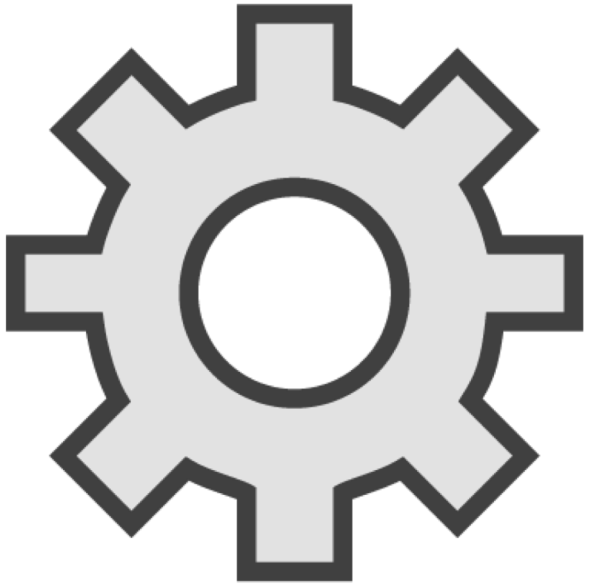
- Character filtering
- Tokenization
- Token filtering

# Demo

- Analyzer API

# Index the data

# Indexing Breakdown

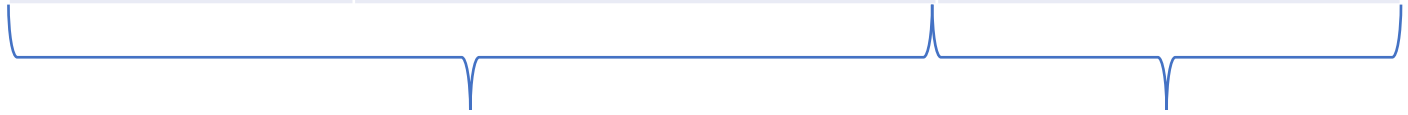


- Tokens and meta-data are stored
- Inverted index

# Inverted Index

| Document ID | Text |
|---|---|
| 1 | Natural coconut water |
| 2 | Natural Mineral Water |

| Term | Freq | Document IDs |
|---|---|---|
| coconut | 1 | 1 |
| mineral | 1 | 2 |
| natural | 2 | 1,2 |
| water | 2 | 1,2 |

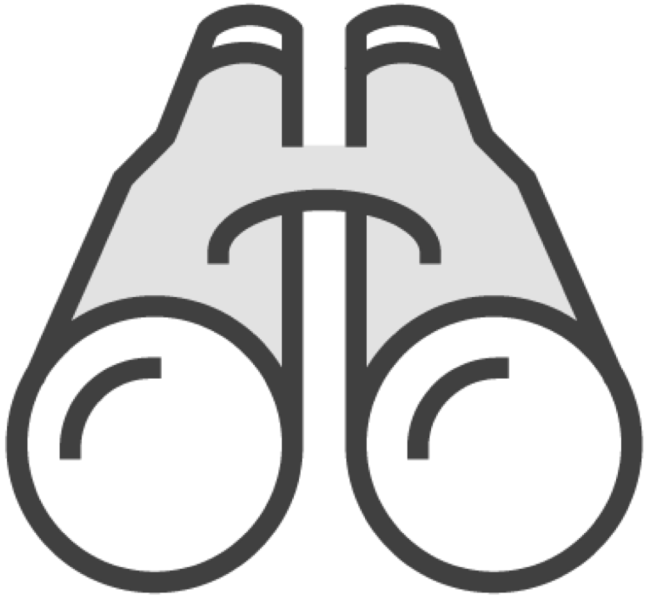


**Dictionary** **Postings**

# Demo

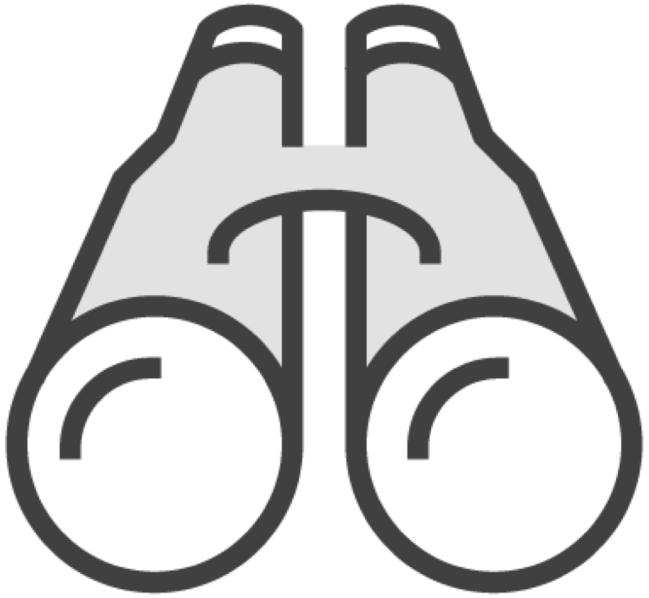
- Boolean search
- Function Score search

# Elasticsearch Queries



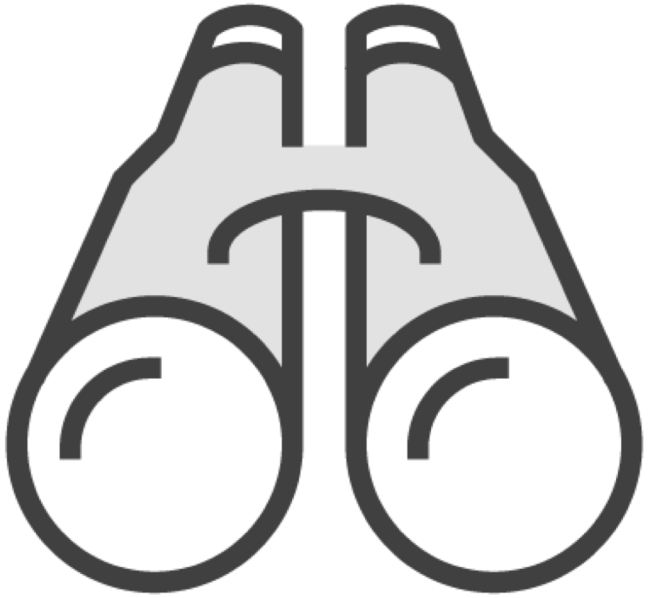
- Relevancy search
- Boolean search

# Elasticsearch Queries – Boolean



- Must
- Should
- Must not
- Combine queries

# Elasticsearch Queries – Function Score

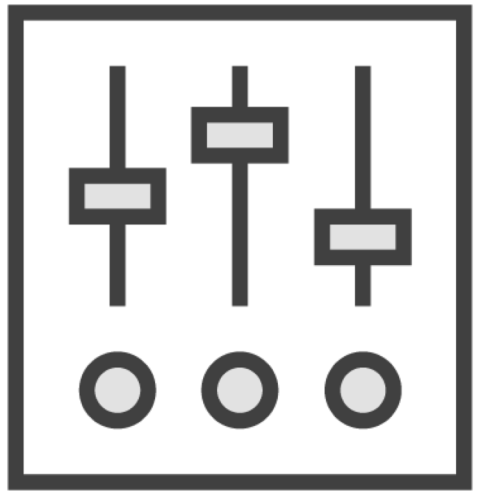


- Boost the rating function of a query
- Filters
- Weights
- Term Frequency
- Inverse Document Frequency

# Demo

- Aggregation

# Elasticsearch – Aggregations



- Support building summaries of data
- Refinements
- Performance considerations

# Demo

- Autocomplete

# Additional Lessons Learned


- Clear understanding of expectations
- Search is an evolving feature
- Real-time processing
- Flatten the index
- Use a template library to build queries
- Breaking changes

# Thank You!