Elastic & food fact checking

Géraud Dugé de Bernonville

30/03/2020

Outline

Qualité des aliments & sécurité sanitaire

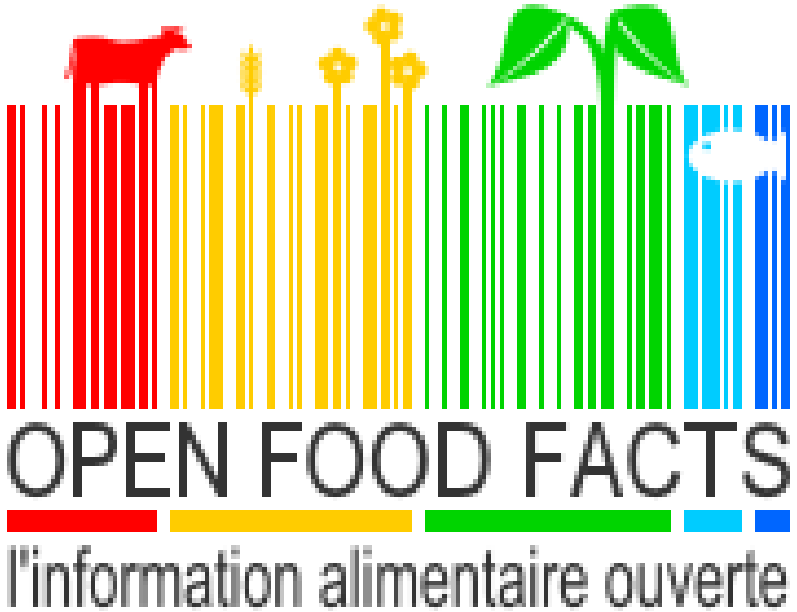
- Vache folle
- Grippe aviaire
- Perturbateurs endocriniens (pesticides, plastiques et autres substances chimiques...)
- OGM
- Allergènes (gluten, crustacés, oeufs, arachides, soja, ...)
- Cancérogènes (E171 - oxyde de titane ?)

Questions :

- Où trouve-t-on ces éléments ?
- Quelles catégories de produit sont les plus concernées ?
- Quelles marques ?

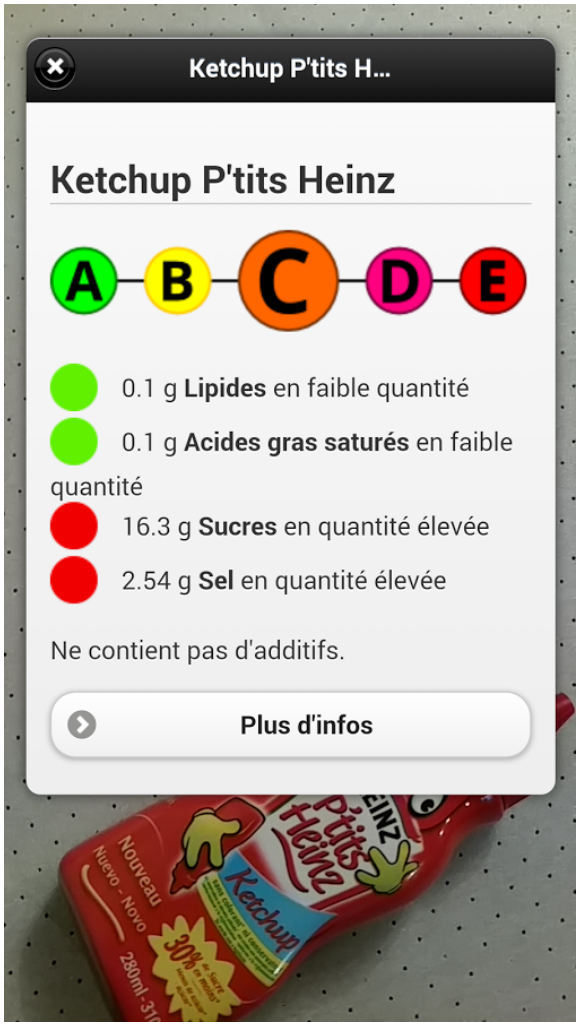
Mais surtout… Y a t'il du E171 dans la bière ?

Open Food Facts



Base de données sur les produits alimentaires faite par tout le monde, pour tout le monde.

Open Food Facts - Mobile



acide d'ammonium, diphosphate disodique, carbonate acide de sodium), sel, lactose et protéines de lait.

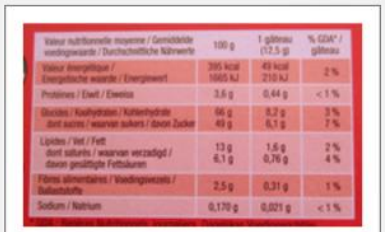
Traces éventuelles : Sésame

Additifs :

- E331 - Citrates de sodium
- E333 - Citrates de calcium
- E330 - Acide citrique
- E440 - Pectines
- E415 - Gomme xanthane
- E322 - Lécithines
- E503 - Carbonates d'ammonium
- E450 - Diphosphate disodique
- E500 - Carbonates de sodium

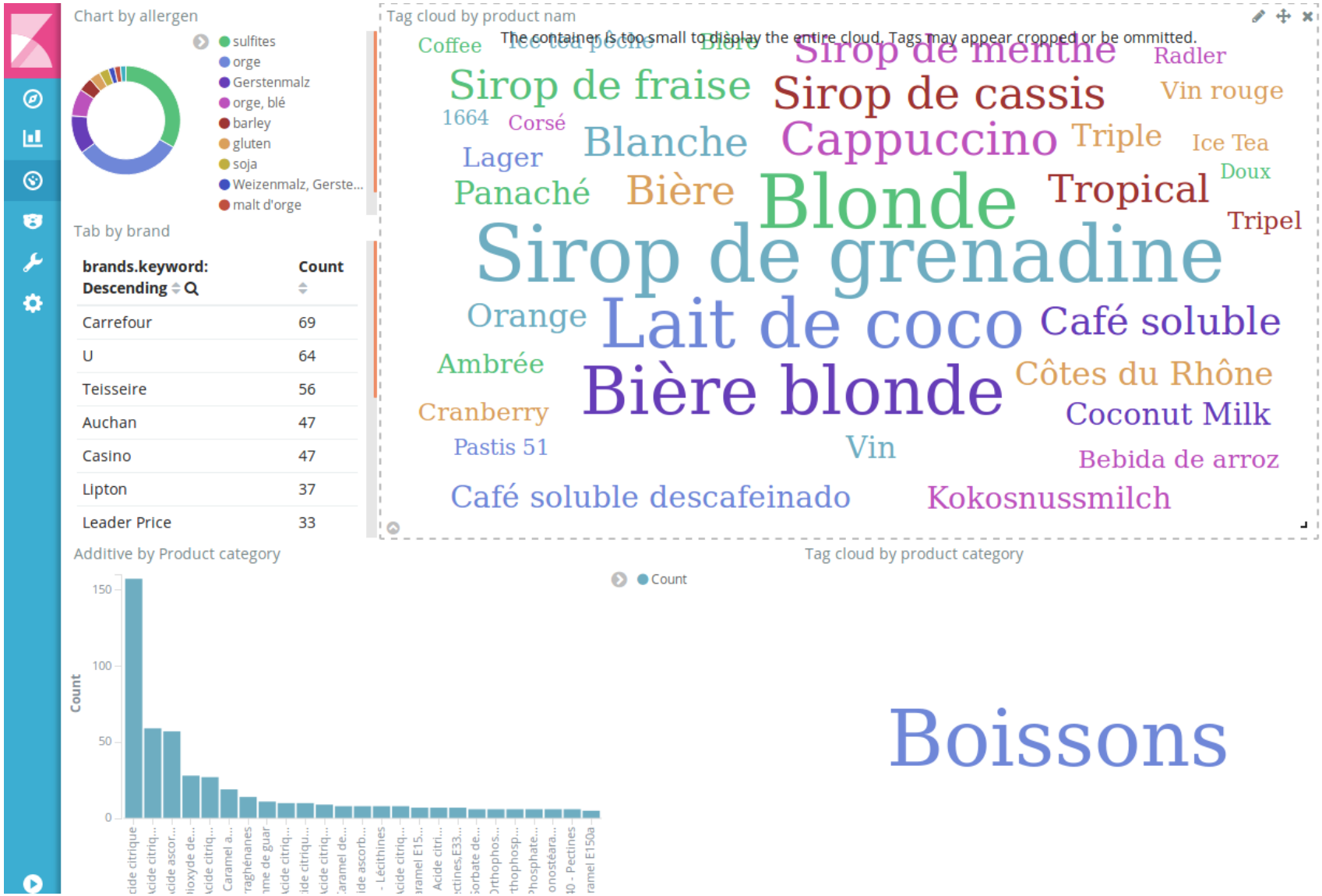
Informations nutritionnelles

Taille d'une portion : 1 gâteau (12,5g)



Informations nutritionnelles

Ce qu'on aimerait avoir



Elastic Stack

- Moteur de recherche
- Analyse et stockage de données

- Ingestion des données

- Visualisation

Topo Elasticsearch

Document JSON

```
{
  "name": "Chips au vinaigre",
  "category" : "apero",
  "lipides" : 20,
  "glucides" : 10,
  "proteines" : 5
}
```

API REST

```
<GET|POST|PUT|DELETE>
http[s]://<hostname>:<port>/[<index>]/[<type>]/[_<keyword>]]
```

- index
- type: _doc
- _keyword : _search, _mapping,...

Installation

Pré-requis

- Docker et docker-compose

Version 7.x

1. `cd elk-compose`
2. Lancer la stack: `docker-compose up`
3. Ouvrir `http://localhost:5601`
4. Aller dans Dev Tools
   (`http://localhost:5601/app/kibana#/dev_tools/console`)

Jouons avec Elasticsearch

Indexer un document

```
POST /store/_doc
{
  "name": "Chips au vinaigre",
  "category" : "apero",
  "lipides" : 20,
  "glucides" : 10,
  "proteines" : 5
}

POST /store/_doc
{
  "name": "Langues piquantes",
  "category" : "confiserie",
  "lipides" : 0,
  "glucides" : 90,
  "proteines" : 5
}
```

Requêter

```
GET /store/_search

GET /store/_search?q=langues

GET /store/_search
{
  "query": {
    "match": {
      "name": "langues"
    }
  }
}
```

Topo Logstash

Lancement

```
docker-compose run logstash -f /pipeline/demo/ingest.conf
```

Fichier conf

```
input { ... }
filter { ... }
output { ... }
```

Jouons avec Logstash - Données de test

1. Récupérer le fichier CSV `sample-fr.openfoodfacts.org.products.csv`
2. Vérifier le fichier `file-input.conf` dans le répertoire pipelines/student

```
input {
  file {
    path => "/data/*.csv"
    start_position => "beginning"
    sincedb_path => "/data/sincedb"
  }
}
```

3. Vérifier le fichier `debug-output.conf`

```
output {
  stdout { codec => "rubydebug" }
}
```

4. Lancer logstash

```
docker-compose run logstash \
       -f /pipelines/student/{file-input,debug-output}.conf
```

5. Patienter...

Ajout du filtre CSV

1. Vérifier le fichier `filter.conf` dans le répertoire pipelines/student

```
filter {
    csv {
        separator => "	"
        autogenerate_column_names => false
        autodetect_column_names => true
    }
    mutate {
        split => { "allergens" => "," }
        split => { "packaging_tags" => "," }
```

2. Supprimer le fichier `since_db`
3. Lancer logstash

```
docker-compose run logstash -w 1 \
      -f '/pipelines/student/{file-input,debug-output,filter}.conf'
```

Ajout de la sortie Elasticsearch

1. Vérifier le fichier `elastic-output.conf`

```
output {
    elasticsearch {
        hosts => [ "elasticsearch" ]
        index => "openfoodfacts"
        template => "/pipelines/student/openfoodfacts.template.json"
        template_name => "openfoodfacts"
        template_overwrite => true
    }
}
```

2. Lancer logstash

```
docker-compose run logstash -w 1 \
      -f '/pipelines/student/{file-input,filter,elastic-output}.conf'
```

Dans Kibana > Dev Tools

```
GET /openfoodfacts/_search
GET /openfoodfacts/_search?q=e171
```

Query time !

Nombre de catégories:

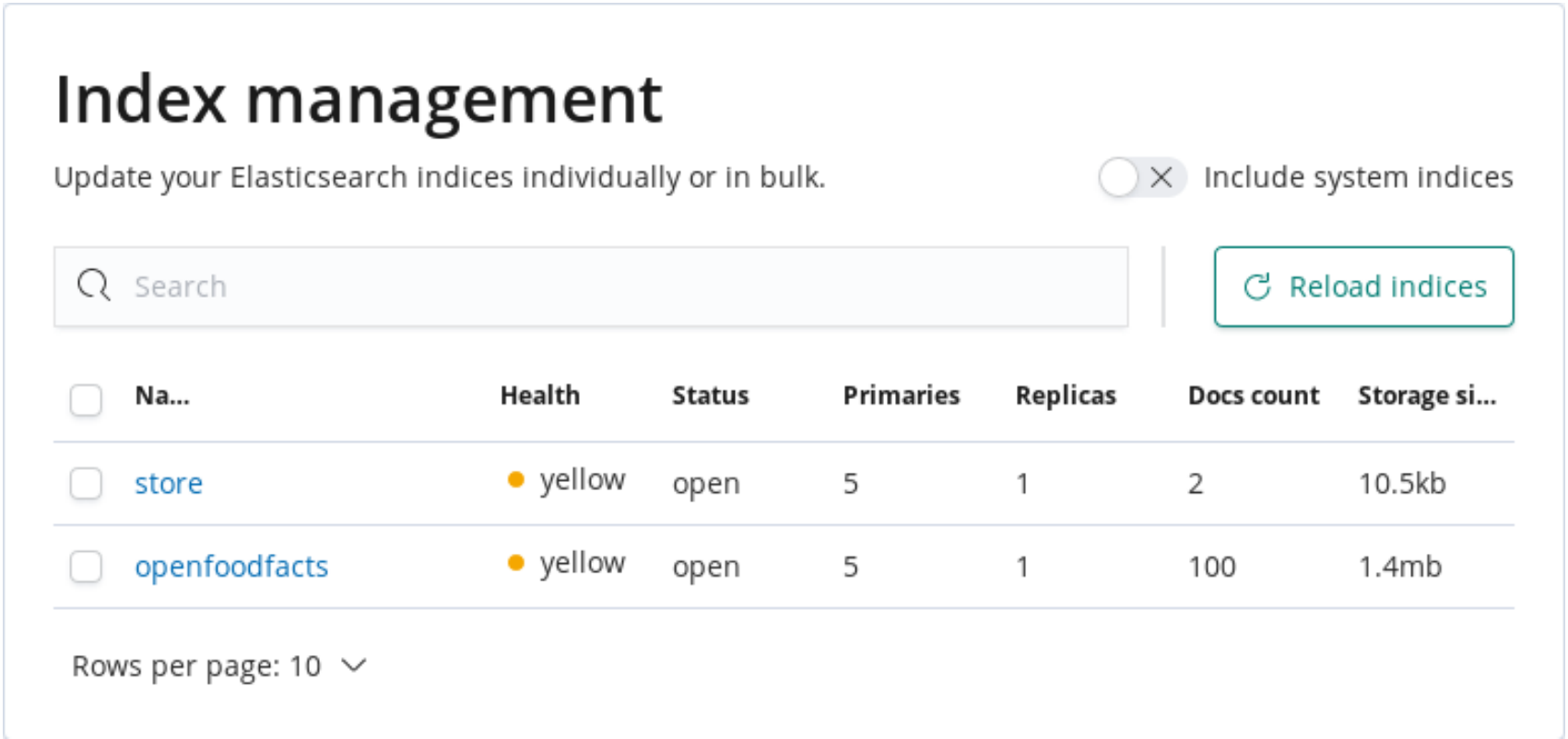
```
GET /openfoodfacts/_search
{
  "aggs": {
    "categories_count": {
      "value_count": {
        "field": "main_category"
      }
    }
  }
}
```

Query time !

Répartition des additifs par catégories:

```
GET /openfoodfacts/_search
{
  "aggs": {
    "par_categorie": {
      "terms": {
        "field": "main_category_fr",
        "size": 10
      },
      "aggs": {
        "par_additif": {
          "terms": {
            "field": "additives_fr"
          }
        }
      }
    }
  }
}
```

Problème de taille



Configuration du mapping

```
DELETE openfoodfacts

PUT /openfoodfacts
{
    "settings" : {
        "number_of_shards": 3,
        "number_of_replicas": 0
    },
    "mappings": {
        "dynamic_templates": [
            {
                "strings": {
                    "match_mapping_type": "string",
                    "mapping": {
                        "type": "keyword"
                    }
                }
            }
        ]
    }
}
```

Jouons avec Kibana

Navigation dans les données

1. Configurer l'index, décocher Index contains time-based events
2. Accéder à l'onglet Discover
3. Sélectionner les champs `additives_fr`, `main_category_fr`,...

Première visualisation - Nuage des principales catégories

1. Accéder à l'onglet Visualize
2. Sélectionner Tag Cloud
3. Configurer un bucket Tags
   - Aggregation = Terms
   - Field = `main_category_fr`
   - Size = 50
   - Custom Label = Catégories principales
4. Sauvegarder le widget

Kibana - Suite

Tableau des marques

1. Sélectionner Table
2. Créer un bucket Split Rows
   - Aggregation = Terms
   - Field = `brands`
   - Size = 20
   - Custom Label = Marques
3. Sauvegarder

Kibana - Mmmmm Donut

Donut des allergènes

1. Sélectionner Pie chart
2. Créer un bucket Split Slices
   - Aggregation = Terms
   - Field = `allergens`
   - Size = 10
   - Custom Label = Allergènes
   - Options > Sélectionner Donut
3. Sauvegarder

Kibana - Fin (?)

Histogramme des additifs

1. Sélectionner Vertical Bar Chart
2. À vous de jouer…

Tag cloud des produits

On veut ça:

Dashboard

1. Ajouter tous les widgets dans un nouveau dashboard
2. Sauvegarder

Chargeons toute la base !

- L'objectif est de voir le résultat avec l'ensemble des données
- Pour éviter les doublons, on supprime l'index `openfoodfacts`
- Décompresser ensuite le fichier `fr.openfoodfacts.org.products.csv.gz` dans votre répertoire `data`
- Lancer logstash

Beer

Mission accomplie !

- Requêtes avec Elasticsearch
- Ingestion de données avec Logstash
- Visualisation avec Kibana

Pour aller plus loin

- Fixer problèmes d'import
  - Champs trop longs
  - Encodage
  - Guillemets mal positionnés
- Découper les champs, par exemple :
  - E330 - Acide citrique,E150c - Caramel ammoniacal,E300 - Acide ascorbique
  - Frais,Produits laitiers,Desserts,Fromages,Fromages blancs,Fromages-blancs-aromatises
- Configurer l'analyseur pour utiliser la langue française
- Utiliser les informations de géolocalisation

Merci

?

Questions