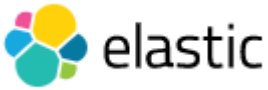
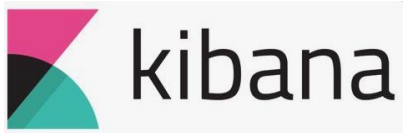
ElasticSearch





Mme Ines Slimene
Ines_slimen@yahoo.fr

Plan module

- Introduction
- Écosystème Hadoop
- HDFS
- MapReduce
- Langages de requête Hadoop :Pig, Hive
- Etude d'un SGBDNR : Hbase
- **Dataviz : ElasticSearch**

Présentation

- Elasticsearch est un outil de recherche distribué en temps réel et un outil d'analyse.
- Il est utilisé pour
  - recherche full text
  - recherche structurée
  - Analyse
- Utilisé par :
  - Wikipedia (http://fr.wikipedia.org)
  - The Guardian (http://www.theguardian.com)
  - StackOverflow (http://stackoverflow.com/)
  - GitHub (https://github.com/)

Dépendances et fonctionnalités

- Elasticsearch a besoin de :
  - Apache Lucene™, un moteur de recherche.
  - Java donc la JVM est requise.
- Elasticsearch est:
  - un stockage de document temps réel distribué où **tous les champs** sont indexés et consultables
  - un moteur de recherche distribué avec de l'analyse temps réel capable de supporter la monté en charge avec une centaine de servers et des peta-octets de données structurées ou non.

Stockage des documents

- Type de stockage : Orienté Document
- Le contenu de chaque Document est indexé
- Un Document possède un Type (qui défini son mapping)
- Les Types sont contenus dans un Index

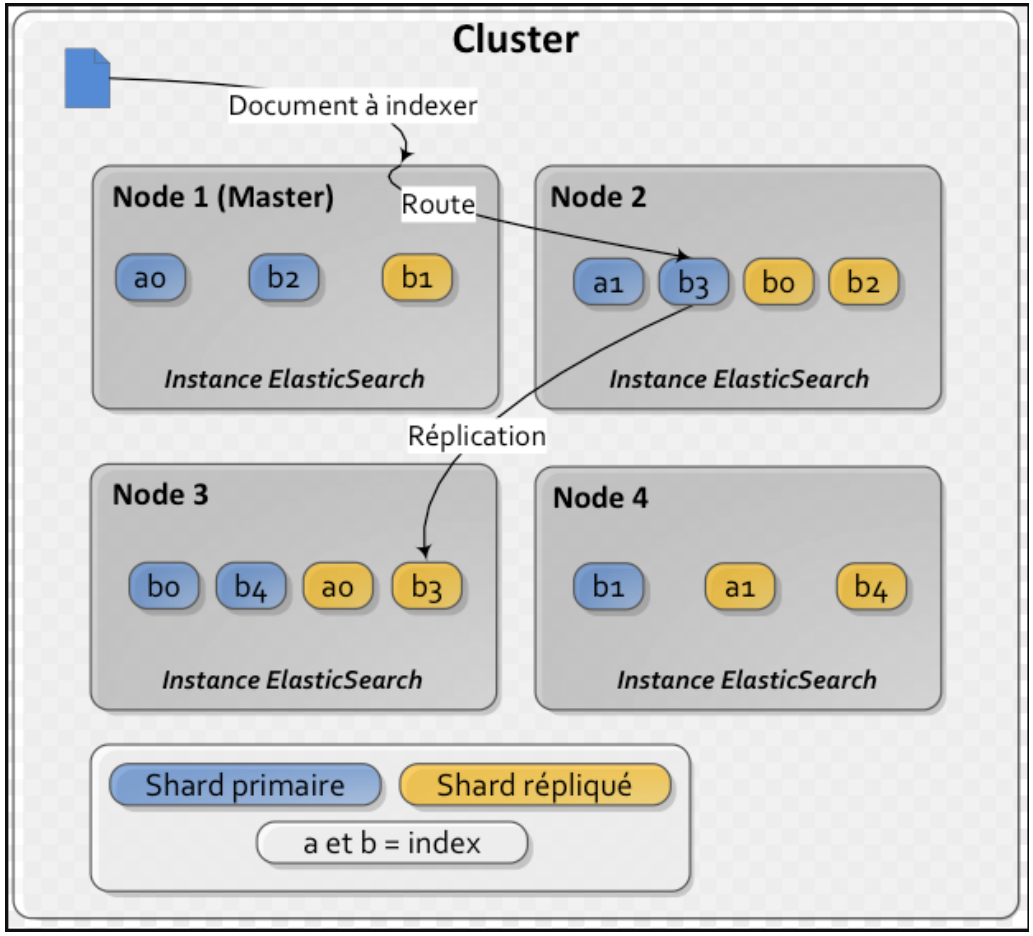
BDR	Elasticsearch
Base de données	Index
Tables	Types
Lignes	Documents
Colonnes	Champs

Architecture

Fonctionnement cluster

- **Nœud** : Correspond à une instance ElasticSearch
- **Cluster** : Il est composé d'un à plusieurs noeuds. Un nœud maître est choisi, il sera remplacé en cas de défaillance.
- **Index** : Espace logique de stockage de documents de même type, découpé sur un à plusieurs **Primary Shards** et peut être répliqué sur zéro ou plusieurs **Secondary Shards.**
- **Shard** : Correspond à une instance Lucène.
- **Primary Shard** : Par défaut, l'index est découpé en 5 Shards Primary. Il n'est pas possible de changer le nombre de partitions après sa création.
- **Secondary Shard** : Il s'agit des partitions répliquées. Il peut en avoir zéro à plusieurs par Primary Shard.

Fonctionnement cluster



Statut du cluster

- Afficher le statut du cluster :

curl -XGET 'http://localhost:9200/_cluster/health?pretty'

- Le champ status donne une indication global sur le fonctionnement du cluster :
  - vert : Tous les Shards primaires et replicas sont actifs (Le cluster fonctionne et la tolérance aux pannes est assurée).
  - jaune : Tous les Shards primaires sont actifs, mais les Shards replicas ne sont pas tous actifs (Le cluster fonctionne mais si un nœud tombe la tolérance aux pannes n'est pas assurée).
  - rouge : Des Shards primaires sont inactifs (Le cluster n'est pas fonctionnel).

Gestion des Shards

- Création d'un index test_shard avec 3 Shards primaires et 1 Shard replica (pour chaque primaire) :

```
curl -XPUT 'http://localhost:9200/test_shard' -d '
{
"settings" : {
"number_of_shards" : 3,
"number_of_replicas" : 1
}
}'
```

- Dans cet état le statut du cluster est "jaune" car les Shards replicas ne peuvent pas être lancés.

Tolérance aux pannes

- 1 Noeud : Un point de défaillance
- La solution est simple : lancer un nouveau Noeud
- Le nouveau Noeud rejoindra automatiquement le cluster s'il a le même nom de cluster (cluster.name).

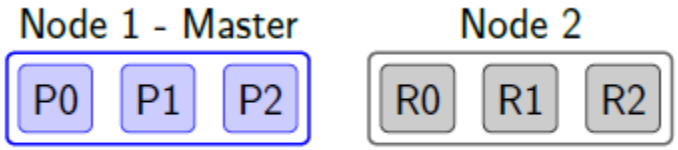


Figure: 2 noeuds avec 3 shards primaires et 1 shard replica pour chaque shard primaire

Communication avec Elasticsearch

Lancement d'Elasticsearch

- curl 'http://localhost:9200/?pretty'

```
[cloudera@quickstart ~]$ curl 'http://localhost:9200/?pretty '
{
  "name" : "i0zfvDj",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "iz1HCCFlTBWG0Z8TVFerhQ",
  "version" : {
    "number" : "5.6.2",
    "build_hash" : "57e20f3",
    "build_date" : "2017-09-23T13:16:45.703Z",
    "build_snapshot" : false,
    "lucene_version" : "6.6.1"
  },
  "tagline" : "You Know, for Search"
}
```

Lancement d'Elasticsearch

- Afficher la liste de tous les indexes

curl 'localhost:9200/_cat/indices?v'

```
[cloudera@quickstart ~]$ curl 'localhost:9200/_cat/indices?v'
health status index               uuid                   pri rep docs.count docs.deleted store.size pri.store.size
yellow open   logstash-2015.05.19 eJ8hH3baQ2q6PLs5DGKp8A   5   1       4624            0     29.1mb         29.1mb
yellow open   logstash-2015.05.20 _9pAiKXSRamspKVYfnz8gg   5   1       4750            0     31.4mb         31.4mb
yellow open   shakespeare         rirh8cZiRlG-DbJy9aJ44Q   5   1     111396            0     28.7mb         28.7mb
yellow open   boutique            gUuH2SxBQiW08c5MEyAuJA   5   1          2            0     10.9kb         10.9kb
yellow open   test_shard          -tZeyRaTSB630xKjdElghw   3   1          0            0       486b           486b
```

API Rest - Type de requête

- POST : création d'un document
- PUT : création ou modification d'un document
- GET : récupération d'un document
- HEAD : test si un document existe
- DELETE : suppression d'un document
- Retourne un code de retour HTTP (200, 404, etc.)
- une réponse encodé en JSON (sauf pour les requêtes HEAD)

Requête PUT : création

- La commande suivante sauvegarde un document dans l'index « boutique" avec comme type « cd" et avec l'id "1" :

http://elastic_search:port/**index**/**type**/**identifiant**

```
curl -XPUT http://localhost:9200/boutique/cd/1 -d '{
"titre" : "19 Nocturnes",
"artiste" : "Arthur Rubinstein",
"compositeur" : "Fryderyk Chopin",
"genre" : "classique"
}'
```

```
["_index":"boutique","_type":"cd","_id":"1","_version":1,"result":"created"
```

Requête PUT : création

- Création automatique de l'ID:

```
curl -XPUT http://localhost:9200/boutique/cd -d '{
"titre" : "19 Nocturnes",
"artiste" : "Arthur Rubinstein",
"compositeur" : "Fryderyk Chopin",
"genre" : "classique"
}'
```

Requête PUT : mise à jour

curl -XPUT http://localhost:9200/boutique/cd/1 -d '{
"titre" : "19 Nocturnes",
"artiste" : "Arthur Rubinstein",
"compositeur" : "Fryderyk Chopin",
"genre" : "classic"}

{"_index":"boutique","_type":"cd","_id":"1","_version":2,"result":"updated"

curl -XPUT http://localhost:9200/boutique/cd/2 -d '{
"titre" : "25",
"artiste" : "Adele Adkins",
"compositeur" : "Greg Kurstin",
"genre" : "Neo soul"}

Requête PUT :

- Réponse :
  - HTTP 200 : Document modifié
  - HTTP 201 : Document créé
  - HTTP 409 : Document existe déjà dans le cas d'une opération de création uniquement (_create)

Requête GET

- Afficher le cd dont l'id est 1

curl -XGET http://localhost:9200/boutique/cd/1?pretty

- Afficher la liste de tous les cds :

curl -XGET http://localhost:9200/boutique/cd/_search?pretty

- Par défaut, la recherche retourne 10 résultats dans le tableau hits.

Requête GET : query et Query DSL

- On peut utiliser l'option *query* (q) pour spécifier un critère de recherche
- Exemple : chercher les documents qui contiennent le mot chopin :

curl -XGET http://localhost:9200/boutique/_search?q=Chopin

- Recherche par *Query DSL* de Elasticsearch :

```
curl -XGET http://localhost:9200/boutique/cd/_search?pretty -d'
{"query":
{"match":
{"titre" : "19 Nocturnes"}
}
}'
```

Requête GET

- Réponse :
  - HTTP 200 : OK
  - HTTP 404 : NOT FOUND

Supprimer un document

- curl -XDELETE http://localhost:9200/boutique/cd/1

{"found":true,"_index":"boutique","_type":"cd","_id":"1","_version":3,"result":"deleted","_shards":{"total":2,"successful":1,"failed":0}}

Mapping

Mapping

- Bien qu'Elasticsearch propose d'appliquer un mapping dynamique par défaut lors de l'indexation de nouveaux documents, il est recommandé de définir comment ces documents doivent être manipulés :
  - La façon dont les documents doivent être analysés et indexés.
  - Les types de données des champs du document.
  - Les relations entre les différents types de documents.
  - La gestion des méta-données du document.
  - La définition de la pertinence par champ/document (boosting)
-

API de mapping

Type	Attribut	Description
String	index	analyzed : pour être indexé, analysé et disponible via la recherche. not_analyzed : recherché mais pas analysé. no : pas recherché.
	analyzer	définit l'analyseur qui sera utilisé lors de l'indexation ou de la recherche.
Nombres	ignore_malformed	ignorer un nombre mal formaté. valeur par défaut : false
	coerce	convertit les strings en nombres. valeur par défaut : false
Date	format	format attendu. exemple : yyy/MM/dd
	ignore_malformed	ignorer les valeurs mal formatées

Mapping : exemple

```
"properties" : {
        "name" : {
          "properties" : {
            "first_name" : {"type" : "string"},
            "last_name" : {"type" : "string"}
          }
        },
        "sid":{"type":"string","index":"not_analyzed"}
      }
    },
    "message" : {"type" : "string"}
```

Types and Mappings

- Afficher le mapping pour le type cd de l'index boutique :

curl -XGET http://localhost:9200/boutique/_mapping/cd?pretty


```
"boutique" : {
  "mappings" : {
    "cd" : {
      "properties" : {
        "artiste" : {
          "type" : "text",
          "fields" : {
            "keyword" : {
              "type" : "keyword",
              "ignore_above" : 256
            }
          }
        },
        "compositeur" : {
          "type" : "text",
          "fields" : {
            "keyword" : {
              "type" : "keyword",
              "ignore_above" : 256
            }
          }
```