



TP1 Mise en place et prise en main de la stack Elastic

Introduction

* Elasticsearch est un serveur de recherche open source avancé basé sur Lucene et écrit en Java.

Il fournit des fonctionnalités de recherche distribuées, en texte intégral ou partiel, basées sur la requête et la géolocalisation, accessibles via une API HTTP REST

* Pour exécuter Elasticsearch, un environnement d'exécution Java (JRE) est requis sur la machine.

```
→ Module4 java -version
java version "1.8.0_202"
Java(TM) SE Runtime Environment (build 1.8.0_202-b08)
Java HotSpot(TM) 64-Bit Server VM (build 25.202-b08, mixed mode)
```

ElasticStack

- **Elasticsearch**
 - NoSQL search engine
- **Logstash**
 - Data collection pipeline tool
- **Kibana**
 - Data visualization tool



Elasticsearch: Téléchargement et installation

1. Télécharger la version 7.17.7 stable d'Elasticsearch

On considère pour ce TP la version 7.17.7 stable d'Elasticsearch , vous pouvez la récupérer ici:

<https://www.elastic.co/fr/downloads/past-releases/elasticsearch-7-17-7>

RQ: la dernière version d'elasticsearch est ici: <https://www.elastic.co/fr/downloads/elasticsearch>

2. Décompresser le fichier et exécuter le binaire d'elasticsearch

Dans le répertoire de elasticsearch, dénommé ici par \$ELASTIC, on trouve les répertoires suivants après décompression :

- \$ELASTIC/bin : exécutable,
- \$ELASTIC/plugins : extensions,
- \$ELASTIC/config : fichiers de configuration,
- \$ELASTIC/logs : fichiers de journalisation en cas de problèmes.

```
+ elasticsearch-7.17.7 ls
bin config jdk lib LICENSE.txt logs modules NOTICE.txt plugins README.asciidoc
+ elasticsearch-7.17.7 ./bin/elasticsearch
warning: usage of JAVA_HOME is deprecated, use ES_JAVA_HOME
Future versions of Elasticsearch will require Java 11; your Java version from [/opt/java/jdk1.8.0_202/jre] does not meet this requirement. Consider switching to a distribution of Elasticsearch with a bundled JDK. If you are already using a distribution with a bundled JDK, ensure the JAVA_HOME environment variable is not set.
warning: usage of JAVA_HOME is deprecated, use ES_JAVA_HOME
Future versions of Elasticsearch will require Java 11; your Java version from [/opt/java/jdk1.8.0_202/jre] does not meet this requirement. Consider switching to a distribution of Elasticsearch with a bundled JDK. If you are already using a distribution with a bundled JDK, ensure the JAVA_HOME environment variable is not set.
[2022-11-04T16:37:17,722][INFO ][o.e.n.Node ] [simplon] version[7.17.7], pid[21685], build[default/tar/78dcaaa8cee33438b91eca7f5c7f56a70fec9e80/2022-10-17T15:29:54.167373105Z], OS[Linux/4.19.0-22-amd64/amd64], JVM[Oracle Corporation/Java HotSpot(TM) 64-Bit Server VM/1.8.0_202/25.202-b08]
[2022-11-04T16:37:17,729][INFO ][o.e.n.Node ] [simplon] JVM home [/opt/java/jdk1.8.0_202/jre], using bundled JDK [false]
[2022-11-04T16:37:17,730][INFO ][o.e.n.Node ] [simplon] JVM arguments [-Xshare:auto, -Des.networkaddress.cache.ttl=60, -Des.networkaddress.cache.negative.ttl=10, -XX:+AlwaysPreTouch, -Xss1m, -Djava.awt.headless=true, -Dfile.encoding=UTF-8, -Djna.nosys=true, -XX:-OmitStackTraceInFastThrow, -Dio.netty.noUnsafe=true, -Dio.netty.noKeySetOptimization=true, -Dio.netty.recycler.maxCapacityPerThread=0, -Dio.netty allocator.numDirectArenas=0, -Dlog4j.shutdownHookEnabled=false, -Dlog4j2.disable.jmx=true, -Dlog4j2.formatMsgNoLookups=true, -Djava.locale.providers=SPI,JRE, -XX:+UseConcMarkSweepGC, -XX:CMSInitiatingOccupancyFraction=75, -XX:+UseCMSInitiatingOccupancyOnly, -Djava.io.tmpdir=/tmp/elasticsearch-5780459412008301599, -XX:+HeapDumpOnOutOfMemoryError, -XX:HeapDumpPath=data, -XX:ErrorFile=logs/hs_err_pid%p.log, -XX:+PrintGCDetails, -XX:+PrintGCDateStamps, -XX:+PrintTenuringDistribution, -XX:+PrintGCApplicationStoppedTime, -Xloggc:logs/gc.log, -XX:+UseGCLogFileRotation, -XX:NumberOfGCLogFiles=32, -XX:GCLogFileSize=64m, -Xms1024m, -Xmx1024m, -XX:MaxDirectMemorySize=536870912, -Des.path.home=/home/simplon/Bureau/Module4/elasticsearch-7.17.7, -Des.path.conf=/home/simplon/Bureau/Module4/elasticsearch-7.17.7/config, -Des.distribution.flavor=default, -Des.distribution.type=tar, -Des.bundled_jdk=true]
[2022-11-04T16:37:21,552][INFO ][o.e.p.PluginsService ] [simplon] loaded module [aggs-matrix-stats]
```

Elasticsearch: Lancer le serveur

3. Dans une console, lancer le serveur :

`$ELASTIC/bin/elasticsearch`

Ne pas éteindre ce serveur, ni fermer la fenêtre !

3.1 Pour vérifier l'état,

3.1.1 ouvrir dans un navigateur : `http://localhost:9200/?pretty`

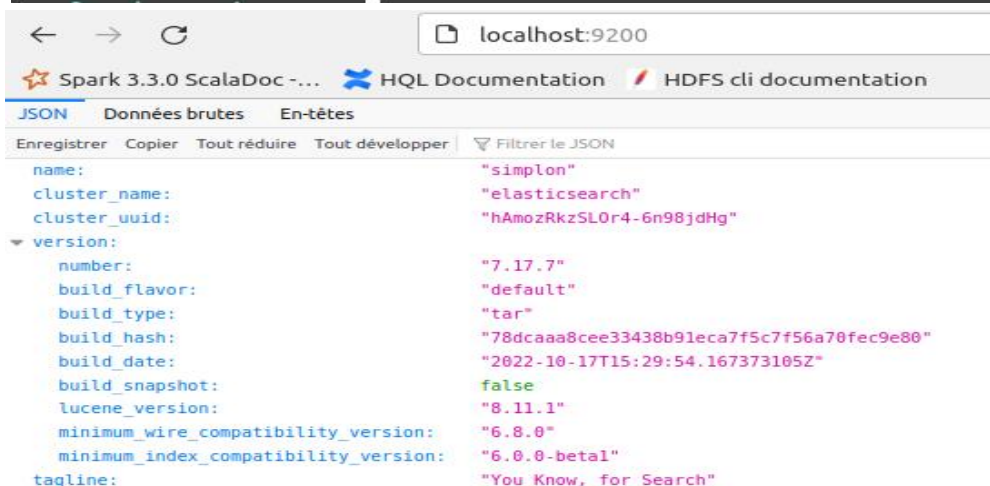
3.1.2 Exécuter la commande CURL (partir de votre navigateur ou d'un client REST pour vérifier si Elasticsearch a été correctement installé.

3.2 Pour éteindre le serveur :

`curl -XPOST 'http://localhost:9200/_shutdown'`

(Ou ctrl C dans le terminal)

```
elasticsearch-7.17.7 curl -X GET http://localhost:9200
{
  "name" : "simplon",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "hAmozRkzSL0r4-6n98jdHg",
  "version" : {
    "number" : "7.17.7",
    "build_flavor" : "default",
    "build_type" : "tar",
    "build_hash" : "78dcaaa8cee33438b91eca7f5c7f56a70fec9e80",
    "build_date" : "2022-10-17T15:29:54.167373105Z",
    "build_snapshot" : false,
    "lucene_version" : "8.11.1",
    "minimum_wire_compatibility_version" : "6.8.0",
    "minimum_index_compatibility_version" : "6.0.0-beta1"
  },
  "tagline" : "You Know, for Search"
}
```



localhost:9200	
Spark 3.3.0 ScalaDoc -... HQL Documentation HDFS cli documentation	
JSON Données brutes En-têtes	
Enregistrer Copier Tout réduire Tout développer Filtre le JSON	
name:	"simplon"
cluster_name:	"elasticsearch"
cluster_uuid:	"hAmozRkzSL0r4-6n98jdHg"
version:	
number:	"7.17.7"
build_flavor:	"default"
build_type:	"tar"
build_hash:	"78dcaaa8cee33438b91eca7f5c7f56a70fec9e80"
build_date:	"2022-10-17T15:29:54.167373105Z"
build_snapshot:	false
lucene_version:	"8.11.1"
minimum_wire_compatibility_version:	"6.8.0"
minimum_index_compatibility_version:	"6.0.0-beta1"
tagline:	"You Know, for Search"

Elasticsearch: Configuration

Pour configurer un serveur, on s'appuie sur le fichier suivant : `$ELASTIC/config/elasticsearch.yml`

- `cluster.name` : nom du cluster pour l'ensemble des noeuds elastic,
- `node.name` : nom du noeud que vous souhaitez démarrer (doit être unique pour un cluster),
- `index.number_of_shards` : nombre de serveurs (défaut 1),
- `index.number_of_replicas` : nombre de serveurs de réplication pour la tolérance aux pannes (défaut 0).

4. Pour faciliter l'administration, télécharger cerebro(web admin tool for elasticsearch)

<https://github.com/lmenezes/cerebro>

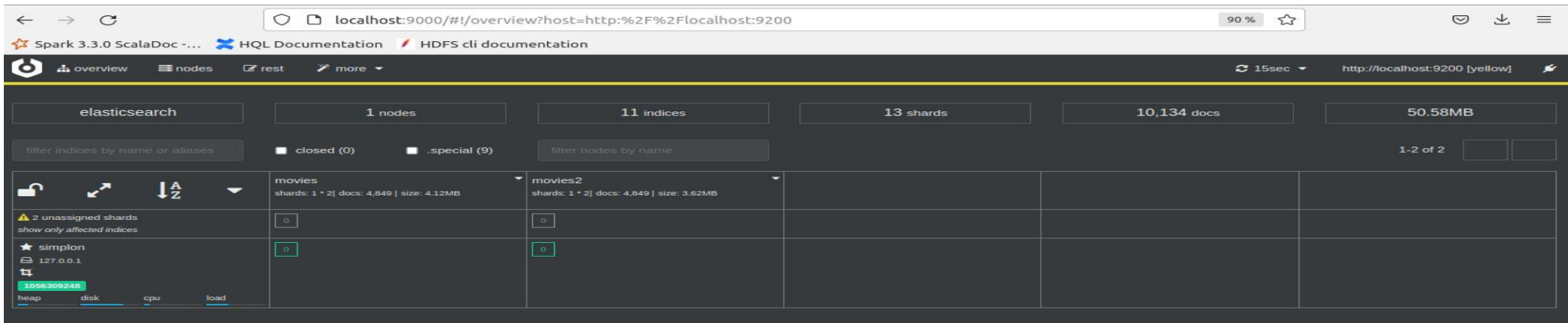
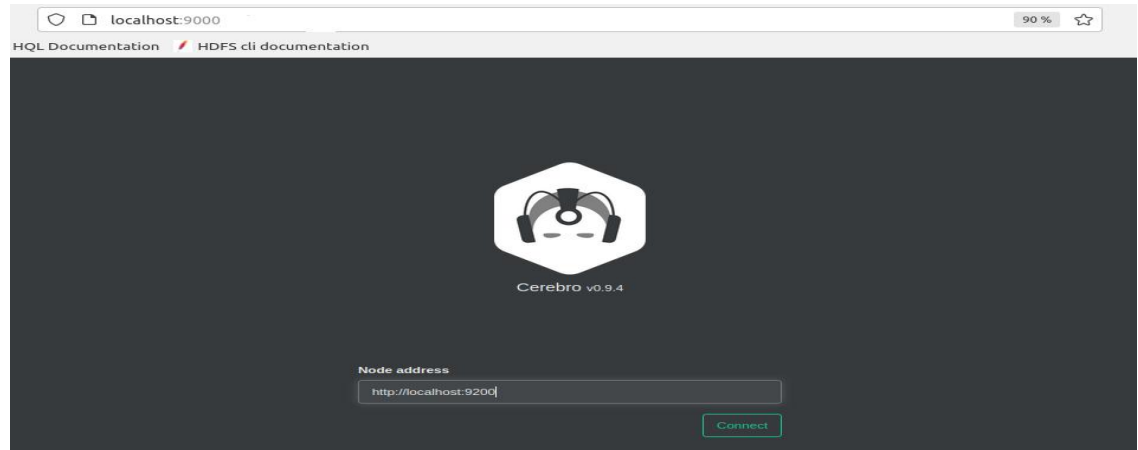
lancer l'interface Web dans une console.

Installation

- Download from <https://github.com/lmenezes/cerebro/releases>
- Extract files
- Run `bin/cerebro`(or `bin/cerebro.bat` if on Windows)
- Access on <http://localhost:9000>

Elasticsearch: Configuration

4.1 Pour ouvrir l'interface cerebro
d'administration : <http://localhost:9000>



Kibana

1. Télécharger la version 7.17.7 stable de Kibana

vous pouvez la récupérer ici: <https://www.elastic.co/fr/downloads/past-releases/kibana-7-17-7>

RQ: la dernière version d'elasticsearch est ici: <https://www.elastic.co/fr/downloads/kibana>

2. Décompresser le fichier et exécuter le binaire de kibana

3. Dans une console, lancer kibana : `$KIBANA/bin/kibana`

Ne pas fermer la fenêtre !

=> Pour vérifier l'état, ouvrir dans un navigateur : <http://localhost:5601>

Elasticsearch

- * Elasticsearch est accessible via une API HTTP REST, généralement via la bibliothèque cURL.

RQ: cURL est un petit exécutable qui envoie/reçoit des requêtes HTTP en ligne de commande. Vous pouvez spécifier des en-têtes en ajoutant à votre commande (XPUT/XGET/XPOST, -H"Content-Type... », -- databinary...).

Il est installé nativement sur Linux, MacOSX et éventuellement sur Windows sinon à télécharger (<https://curl.se/dlwiz/?type=bin>)

- * Les messages entre le serveur de recherche et le client (ou votre application) sont envoyés sous la forme de chaînes JSON.

- * Par défaut, Elasticsearch s'exécute sur le port 9200. (<http://localhost:9200>)

- * On peut nommer aussi une base 'elastic' , un **index**, On peut comparer cela à une collection en NoSQL ou une table étendue en relationnel.

=> Ainsi, pour importer des données dans elasticsearch, il faut préciser cet index et ce type en prefixant chaque document importé par : {"index":{"_index": "MA BASE", "_id":1}}

L'identifiant "_id" doit être unique dans le type et sera associé au document suivant.

Elasticsearch: Importer les données

Pour importer des données dans Elasticsearch, il y a trois manières de le faire :

- Service bulk avec curl (rapide),
- Interface web avec Kibana (lent mais facile),
- Logstash/Intégrations (automatisation)

Service bulk avec Curl

Cette manière d'importer les données utilise un exécutable (en ligne de commande) permettant de simuler les interactions avec des URLs (dont les API REST). Très utile pour faire des instructions simples.

Pour importer les données :

- Télécharger le dataset movies,
- Décompresser l'archive,
- Exécuter la commande:

```
curl -XPUT localhost:9200/_bulk -H "Content-Type: application/json" --data-binary @movies.json
```

```
Module4 curl -XPUT localhost:9200/_bulk_H "Content-Type: application/json" --data-binary @movies.json
{"took":60004,"errors":true,"items":[{"_index":{"_index":"movies","_type":"movie","_id":"1","status":503,"error":{"type":"unavailable_shards_exception","reason":["movies]
[0] primary shard is not active Timeout: [1m], request: [BulkShardRequest [[movies][0] containing [4999] requests]]}}},{_index":{"_index":"movies","_type":"movie","_i
d":"2","status":503,"error":{"type":"unavailable_shards_exception","reason":["movies]
[0] primary shard is not active Timeout: [1m], request: [BulkShardRequest [[movies][0] containing [4999] requests]]}}},{_index":{"_index":"movies","_type":"movie","_i
d":"3","status":503,"error":{"type":"unavailable_shards_exception","reason":["movies]
[0] primary shard is not active Timeout: [1m], request: [BulkShardRequest [[movies][0] containing [4999] requests]]}}},{_index":{"_index":"movies","_type":"movie","_i
d":"4","status":503,"error":{"type":"unavailable_shards_exception","reason":["movies]
[0] primary shard is not active Timeout: [1m], request: [BulkShardRequest [[movies]
[0] containing [4999] requests]]}}},{_index":{"_index":"movies","_type":"movie","_id":"5","status":503,"error":{"type":"unavailable_shards_exception","reason":["movies]
[0] primary shard is not active Timeout: [1m], request: [BulkShardRequest [[movies][0] containing [4999] requests]]}}},{_index":{"_index":"movies","_type":"movie","_i
d":"6","status":503,"error":{"type":"unavailable_shards_exception","reason":["movies]
[0] primary shard is not active Timeout: [1m], request: [BulkShardRequest [[movies]
[0] containing [4999] requests]]}}},{_index":{"_index":"movies","_type":"movie","_id":"7","status":503,"error":{"type":"unavailable_shards_exception","reason":["movies]
[0] primary shard is not active Timeout: [1m], request: [BulkShardRequest [[movies][0] containing [4999] requests]]}}},{_index":{"_index":"movies","_type":"movie","_i
d":"8","status":503,"error":{"type":"unavailable_shards_exception","reason":["movies]
[0] primary shard is not active Timeout: [1m], request: [BulkShardRequest [[movies]
[0] containing [4999] requests]]}}},{_index":{"_index":"movies","_type":"movie","_id":"9","status":503,"error":{"type":"unavailable_shards_exception","reason":["movies]
```

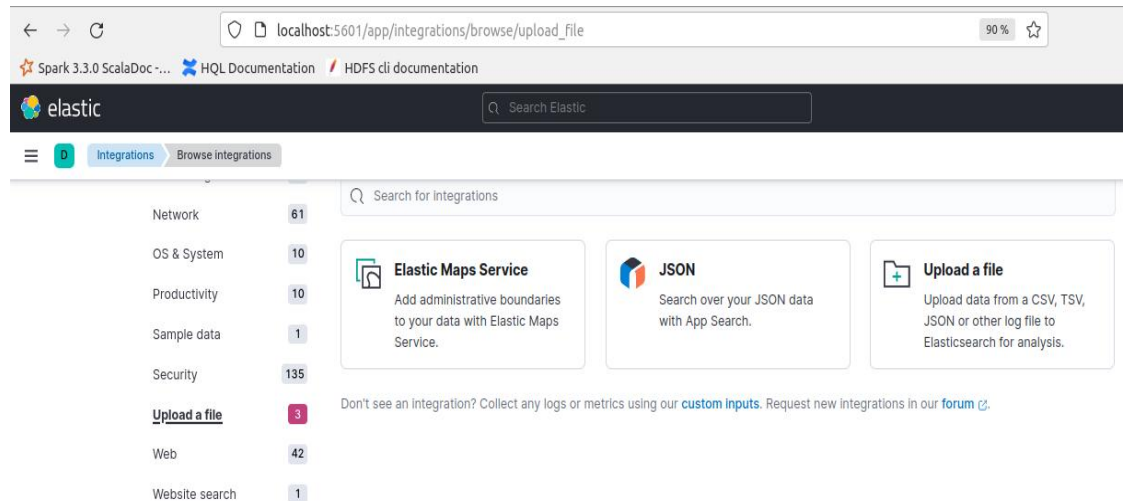
Interface Web de Kibana

Vous pouvez importer votre dataset également en utilisant l'interface de Kibana :

- Aller sur kibana : <http://localhost:5601>
- Aller sur : Menu/Management/Integrations
- Dans la liste des catégories -> Choisir “upload a file”
- Drag and drop le fichier “movies.json”
- Cliquer sur “Import” en bas à gauche.
- Donner un nom à l'index “movies”

=> Cette opération peut prendre du temps car
c'est une interface Web.

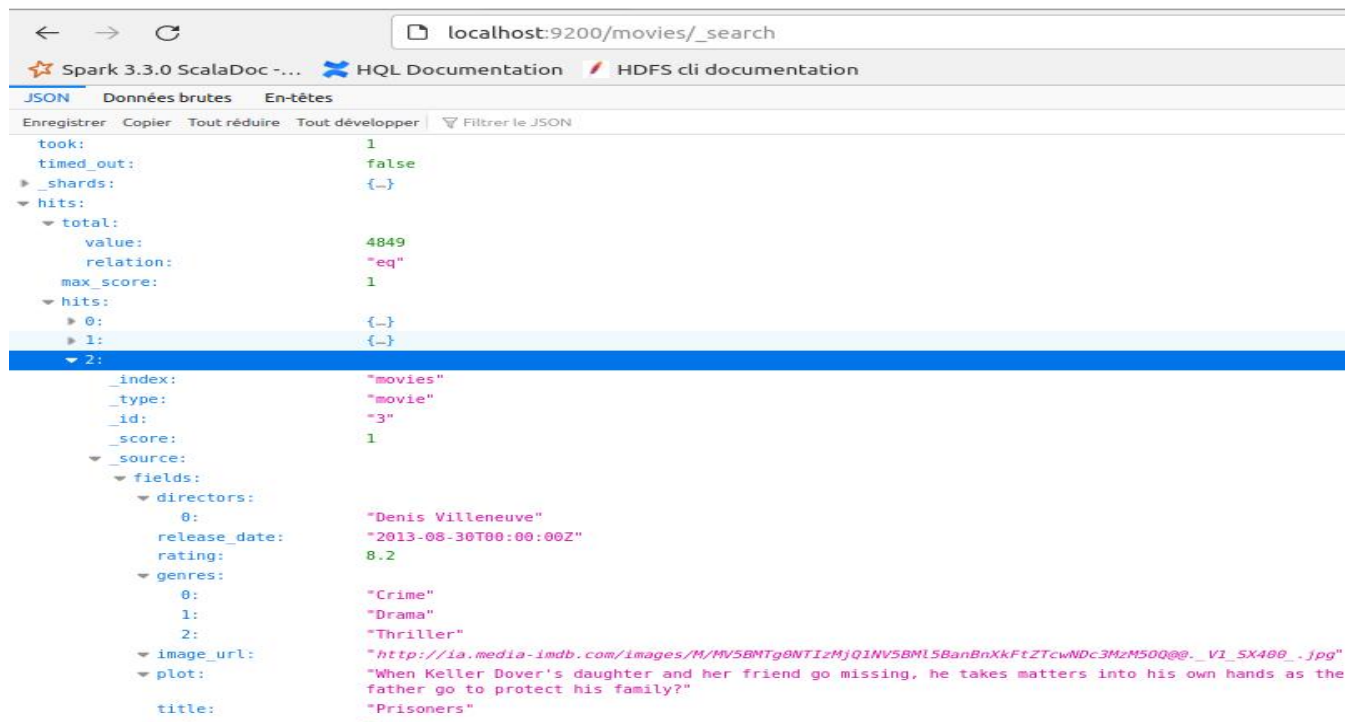
Et oui, il faut préférer curl ou intégrations (comme logstash)



Elasticsearch: Vérifier les données

Pour vérifier des données importées dans Elasticsearch, il y a plusieurs manières de le faire :

1 - via un navigateur en utilisant un webService REST: http://localhost:9200/movies/_search



The screenshot shows a web browser window with the address bar displaying `localhost:9200/movies/_search`. The browser has several tabs open, including "Spark 3.3.0 ScalaDoc -...", "HQL Documentation", and "HDFS cli documentation". The main content area displays the JSON response from the Elasticsearch REST API. The response is a JSON object with the following structure:

```
{
  "took": 1,
  "timed_out": false,
  "_shards": {
    "total": 1,
    "successful": 1,
    "skipped": 0
  },
  "hits": {
    "total": {
      "value": 4849,
      "relation": "eq",
      "max_score": 1
    },
    "hits": [
      {
        "_index": "movies",
        "_type": "movie",
        "_id": "3",
        "_score": 1,
        "_source": {
          "fields": {
            "directors": [
              "Denis Villeneuve"
            ],
            "release_date": "2013-08-30T00:00:00Z",
            "rating": 8.2,
            "genres": [
              "Crime",
              "Drama",
              "Thriller"
            ],
            "image_url": "http://ia.media-imdb.com/images/M/MV5BMTg0NTIzMjQ1NV5BMl5BanBnXkFtZTcwNDc3MzMs50Q@@._V1_SX480_.jpg",
            "plot": "When Keller Dover's daughter and her friend go missing, he takes matters into his own hands as the father go to protect his family?",
            "title": "Prisoners"
          }
        }
      }
    ]
  }
}
```

Elasticsearch: Vérifier les données

Pour vérifier des données importées dans Elasticsearch, il y a plusieurs manières de le faire :

2 - via le terminal avec une commande cURL: `curl -X GET http://@server:numPort/indexName/typeName/1`

```
elasticsearch-7.17.7 curl -X GET http://localhost:9200/movies/_search?pretty
{
  "took" : 1,
  "timed_out" : false,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : {
      "value" : 4849,
      "relation" : "eq"
    },
    "max_score" : 1.0,
    "hits" : [
      {
        "_index" : "movies",
        "_type" : "movie",
        "_id" : "1",
        "score" : 1.0,
        "_source" : {
          "fields" : {
            "directors" : [
              "Joseph Gordon-Levitt"
            ],
            "release date" : "2013-01-18T00:00:00Z",
            "rating" : 7.4,
            "genres" : [
              "Comedy",
              "Drama"
            ],
            "image url" : "http://ia.media-imdb.com/images/M/MV5BMTQxNTc3NDM2MF5BMl5BanBnXkFtZTcwNzQ5NTQ3OQ@@. V1_SX400 .jpg",
            "plot" : "A New Jersey guy dedicated to his family, friends, and church, develops unrealistic expectations from watchi
```

```
elasticsearch-7.17.7 curl -X GET http://localhost:9200/movies/movie/1
{"_index":"movies","_type":"movie","_id":"1","version":1,"seq_no":0,"primary_term":1,"found":true,"source":{"fields":{"directors":["Joseph Gordon-Levitt"],"release date":"2013-01-18T00:00:00Z","rating":7.4,"genres":["Comedy","Drama"],"image url":"http://ia.media-imdb.com/images/M/MV5BMTQxNTc3NDM2MF5BMl5BanBnXkFtZTcwNzQ5NTQ3OQ@@. V1_SX400 .jpg","plot":"A New Jersey guy dedicated to his family, friends, and church, develops unrealistic expectations from watching porn and works to find happiness and intimacy with his potential true love.","title":"Don Jon","rank":1,"running_time_secs":5400,"actors":["Joseph Gordon-Levitt","Scarlett Johansson","Julianne Moore"],"year":2013,"id":"tt2229499","type":"add"}}}
```

Elasticsearch: Vérifier les données

Pour vérifier des données importées dans Elasticsearch, il y a plusieurs manières de le faire :

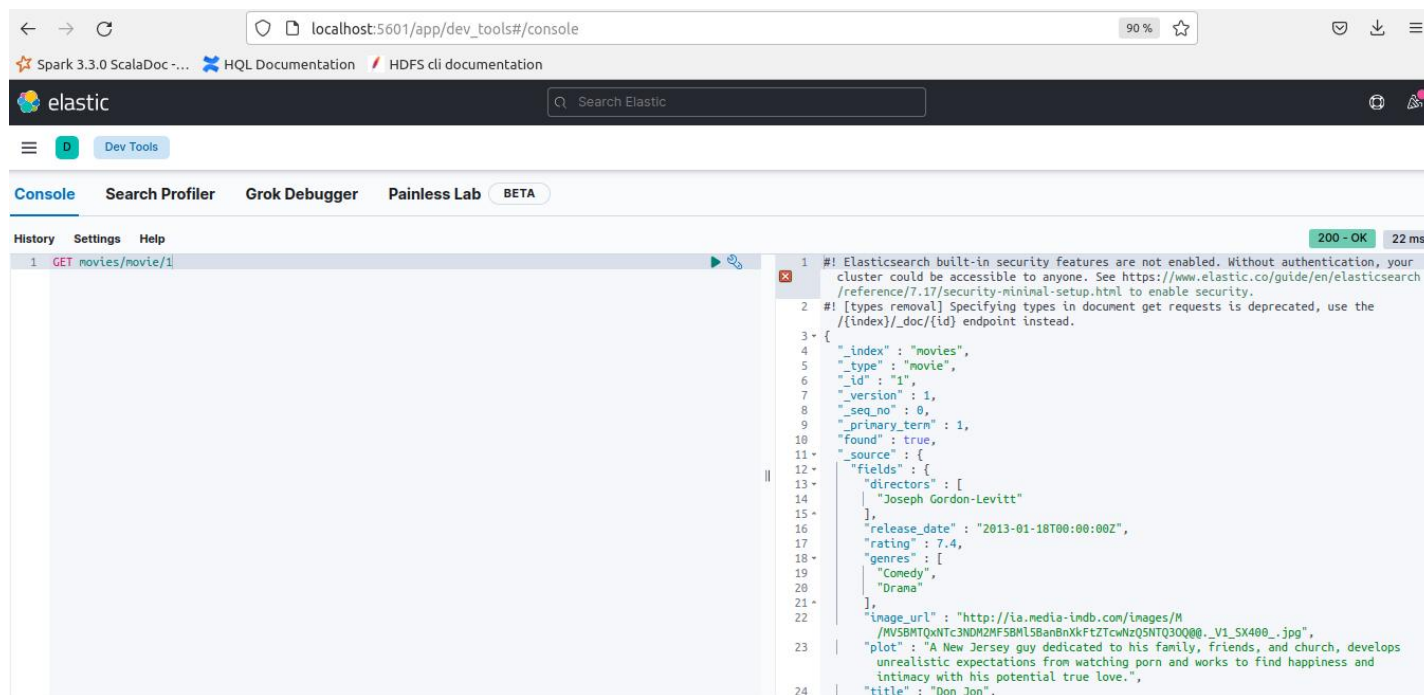
3 - via l'interface web de Kibana devtools

- Aller sur kibana :

<http://localhost:5601>

- Aller ensuite sur :

Menu/Management/Dev Tools



The screenshot shows the Kibana DevTools console interface. The browser address bar indicates the URL is `localhost:5601/app/dev_tools#/console`. The console header shows the 'elastic' logo and a search bar. Below the header, there are tabs for 'Console', 'Search Profiler', 'Grok Debugger', 'Painless Lab', and 'BETA'. The 'Console' tab is active, showing a history of commands. The first command in the history is `GET movies/movie/1`. The console output shows a successful response with a status of `200 - OK` and a response time of `22 ms`. The response body is a JSON object representing a movie document:

```
1 #! Elasticsearch built-in security features are not enabled. Without authentication, your
2 #! cluster could be accessible to anyone. See https://www.elastic.co/guide/en/elasticsearch
3 #! [types removal] Specifying types in document get requests is deprecated, use the
4 #! [{index}/{_doc/{id} endpoint instead.
5 {
6   "_index": "movies",
7   "_type": "movie",
8   "_id": "1",
9   "_version": 1,
10  "_seq_no": 0,
11  "_primary_term": 1,
12  "found": true,
13  "_source": {
14    "fields": {
15      "directors": [
16        "Joseph Gordon-Levitt"
17      ],
18      "release_date": "2013-01-18T00:00:00Z",
19      "rating": 7.4,
20      "genres": [
21        "Comedy",
22        "Drama"
23      ],
24      "image_url": "http://ia.media-imdb.com/images/M
25      /MV5BMTQwNTc3NDM2MzF5SjBhbnBnXkFzZTcwMzQ5NTQ3OQ@@_V1_SX400_.jpg",
26      "plot": "A New Jersey guy dedicated to his family, friends, and church, develops
27      unrealistic expectations from watching porn and works to find happiness and
28      intimacy with his potential true love.",
29      "title": "Don Jon",
```

Elasticsearch: Vérifier les données

Pour vérifier des données importées dans Elasticsearch, il y a plusieurs manières de le faire :

4 - via l'interface web cerebro

- Aller sur kibana :

<http://localhost:9000>

- Aller ensuite sur l'onglet REST (en haut)

