# # Prise en main d’Ansible

## ## Résumé

Depuis plusieurs mois, je dois mettre en place une Chaîne de Déploiement Continue (CDC). Pour cela, je m'appuie sur la technologie [Ansible]. Cette technologie permet d'automatiser des traitements sur un parc de machines. Je me suis donc enfin décidé à réaliser cet article afin de vous expliquer :

* Comment fonctionne Ansible ?
* Comment réaliser son premier playbook ?
* Comment réaliser son premier rôle ?
* Aller plus loin avec Ansible Tower, Ansible Galaxy & co.

Bonne lecture !

## ## Introduction

Comme d'habitude avec moi, je ne me présente pas en expert d'Ansible. L'idée, à travers cet article, est de vous donner quelques conseils pour mieux prendre en main cette technologie.

Cet article a vocation ainsi à :

* Vous permettre de comprendre le fonctionnement d'Ansible
* De prendre en main Ansible à travers votre premier playbook
* De modulariser ses déploiements avec l'utilisation des rôles
* D'aller plus loin dans l'écosystème d'Ansible

Warning : La [documentation officielle] a été mon principal point d'information. Je ne vous saurais donc trop en recommander la lecture. Elle vous permettra d'en apprendre bien plus.

## ## Prérequis

Avant de débuter, je dois vous préciser que ce billet n’a pas vocation à vous expliquer des intérêts ou principes de fonctionnement d’Elasticsearch. Je pars du principe que vous les connaissez ou que cela vous importe peu au vu de vos besoins. Ce billet se centre donc exclusivement sur sa mise en place et son utilisation.

Je préfère également vous préciser la pile applicative que j’utilise afin de mettre en place Elasticsearch :

* Système d’exploitation : Windows 10
* JDK Java : 1.8.0\_201
* Elasticsearch 6.6.0
* Kibana 6.6.0

## ## 1ère étape : Installer la suite Elastic

Afin de prendre en main Elasticsearch, je m’appuie sur plusieurs outils de la suite Elastic. La documentation officielle s’appuie sur la suite :

* Beats
* Elasticsearch
* Elasticsearch Hadoop
* Kibana
* Logstash

Warning : Dans le cadre de la suite Elastic, il est nécessaire d’utiliser la même version pour l’ensemble des produits de la suite.

Pour mon cas pratique, je m’appuie simplement sur Elasticsearch et Kibana en version 6.6.0. Vous trouverez ci-dessous le détail de la procédure d’installation de ces outils.

### ### Comment installer Elasticsearch ?

Elasticsearch est un moteur de recherche utilisant « Lucene » pour l’indexation et la recherche de données.

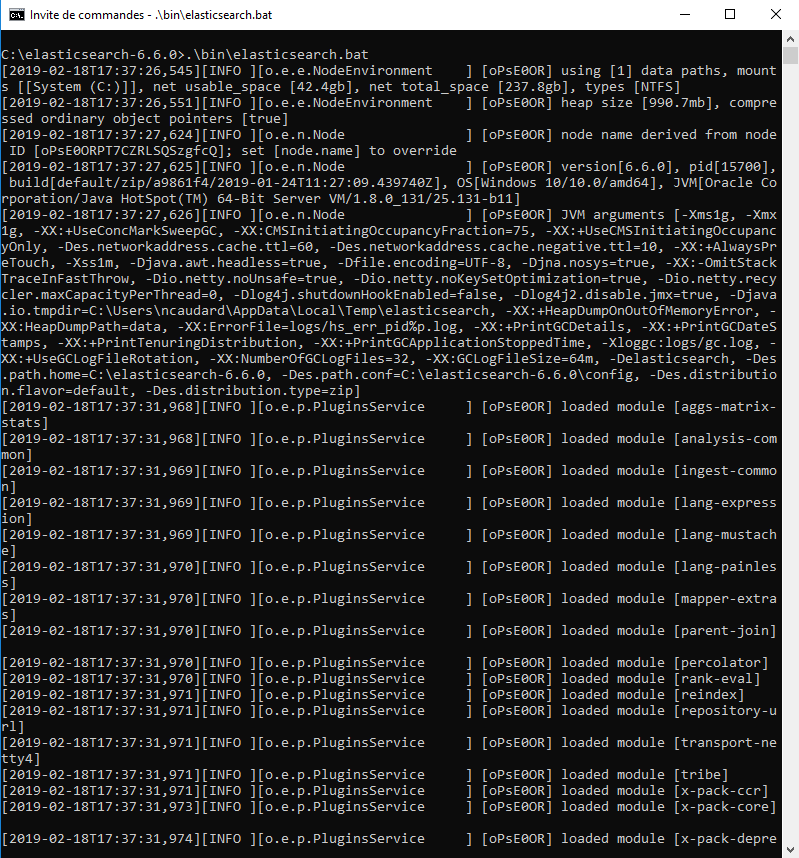
#### #### Installer Elasticsearch sur Windows avec une archive ZIP

Elasticsearch peut s’installer de plusieurs manières : de l’archive, au package Debian en passant par le conteneur Docker ou l’installeur Windows. Dans le cadre de la prise en main d’Elasticsearch, je me suis appuyé sur la [procédure d’installation Windows](https://www.elastic.co/guide/en/elasticsearch/reference/6.6/zip-windows.html).

Pour installer Elasticsearch, vous devez réaliser les actions suivantes :

1. Télécharger l’archive (au format ZIP) d’Elasticsearch V6.6.0 sur le [site officiel](<https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-6.6.0.zip>)
2. Dézipper cette archive au sein du chemin : « C:\elasticsearch-6.6.0 »
3. Ouvrir une invite de commande Windows :

|  |
| --- |
| cd C:\elasticsearch-6.6.0  .\bin\elasticsearch.bat |

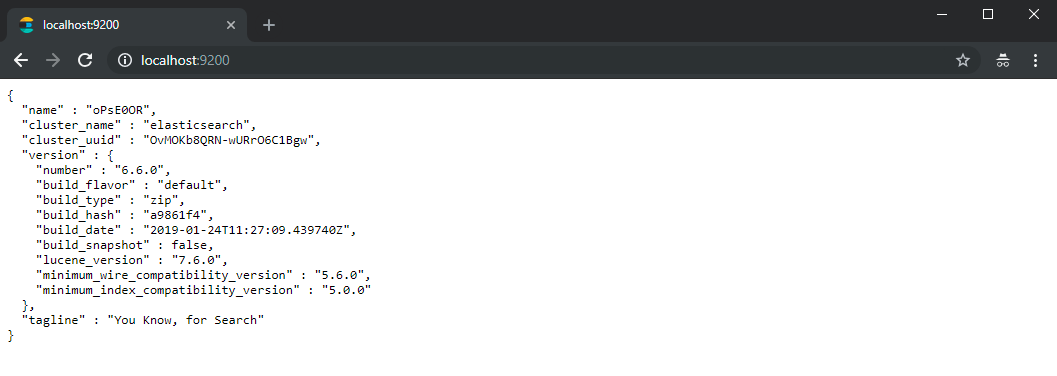


Warning : Par défaut, Elasticsearch trace ses logs au sein de l’invite de commande. Il est alors possible de le couper avec la commande « CTRL + C ».

#### #### Vérifier qu’Elasticsearch fonctionne

Pour vérifier qu’Elasticsearch est correctement installé, ouvrez un navigateur web à l’adresse suivante : <http://localhost:9200/>

Le navigateur devrait alors vous donner l’affichage suivant :



### ### Comment installer Kibana ?

Kibana est un outil de visualisation de données pour Elasticsearch. Il permet la visualisation de contenu indexé au sein d’Elasticsearch.

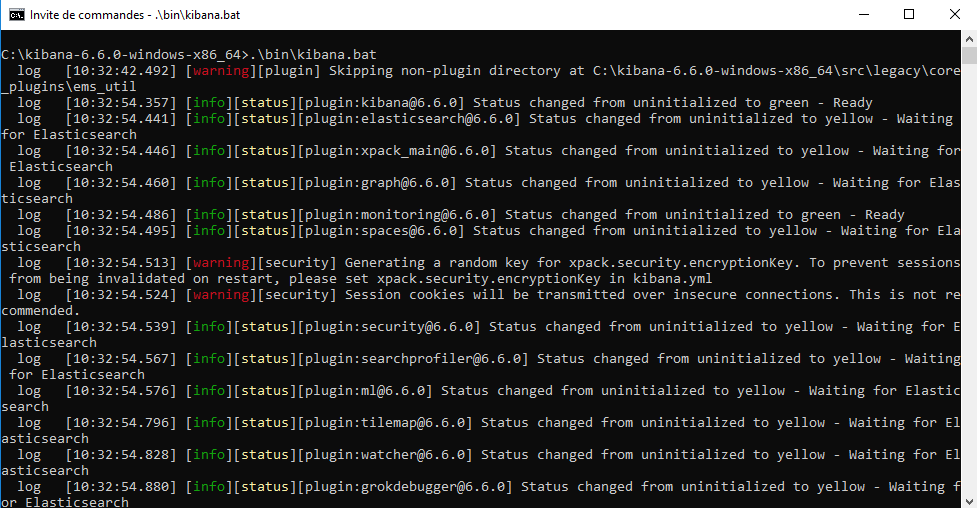
#### #### Installer Kibana sur Windows avec une archive ZIP

Kibana comme Elasticsearch peut s’installer de plusieurs manières : de l’archive au package Debian en passant par le package RPM. Dans le cadre de la prise en main d’Elasticsearch, je me suis appuyé sur la [procédure d’installation Windows](<https://www.elastic.co/guide/en/kibana/6.6/windows.html>).

Pour installer Kibana, vous devez réaliser les actions suivantes :

1. Télécharger l’archive (au format ZIP) de Kibana V6.6.0 sur le [site officiel]( <https://artifacts.elastic.co/downloads/kibana/kibana-6.6.0-windows-x86_64.zip>)
2. Dézipper cette archive au sein du chemin : « C\kibana-6.6.0-windows-x86\_64 »
3. Ouvrir une invite de commande Windows

|  |
| --- |
| cd C:\kibana-6.6.0-windows-x86\_64  .\bin\kibana.bat |

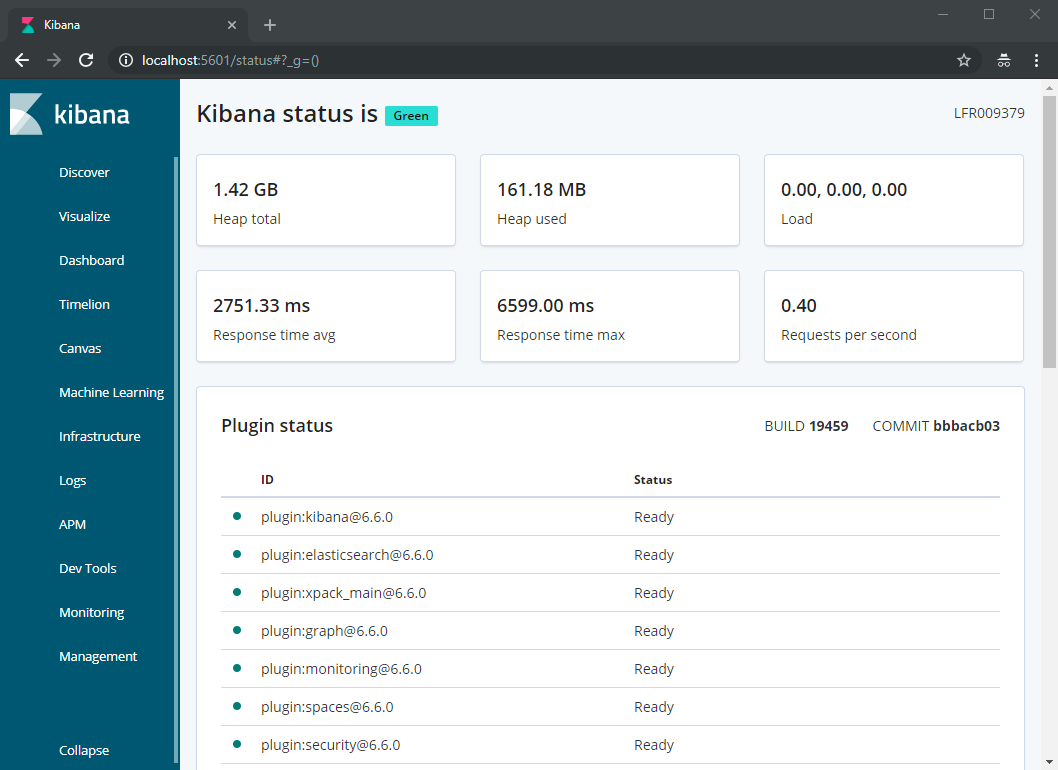


Warning : Par défaut, Kibana trace ses logs au sein de l’invite de commande. Il est alors possible de le couper avec la commande « CTRL + C ».

#### #### Vérifier que Kibana fonctionne

Pour vérifier que Kibana est correctement installé, ouvrez un navigateur web à l’adresse suivante : <http://localhost:5601/status>

Le navigateur devrait alors vous donner l’affichage suivant :



## ## 2ème étape : Tester Elasticsearch avec Kibana

Maintenant que les outils Elasticsearch et Kibana sont correctement installés, nous allons tester le moteur Elasticsearch. Kibana permet de tester les fonctionnalités basiques très facilement.

Warning : Au sein d’Elasticsearch, les concepts suivants sont manipulés : Cluster, Node, Index et Document. Dans la suite de cet article, nous utiliserons essentiellement les mentions Index et Document.

* Un index est une collection de documents qui disposent de caractéristiques similaires
* Un document est une unité basique d’information qui peut être indexé

Dans notre cadre de test, nous allons partir du scénario suivant : « à travers Elasticsearch, nous allons y stocker une liste de documents. Nous nous appuierons sur Elasticsearch pour rechercher les documents les plus appropriés au vu de mes critères de recherche. Chaque document est constitué des informations suivantes :

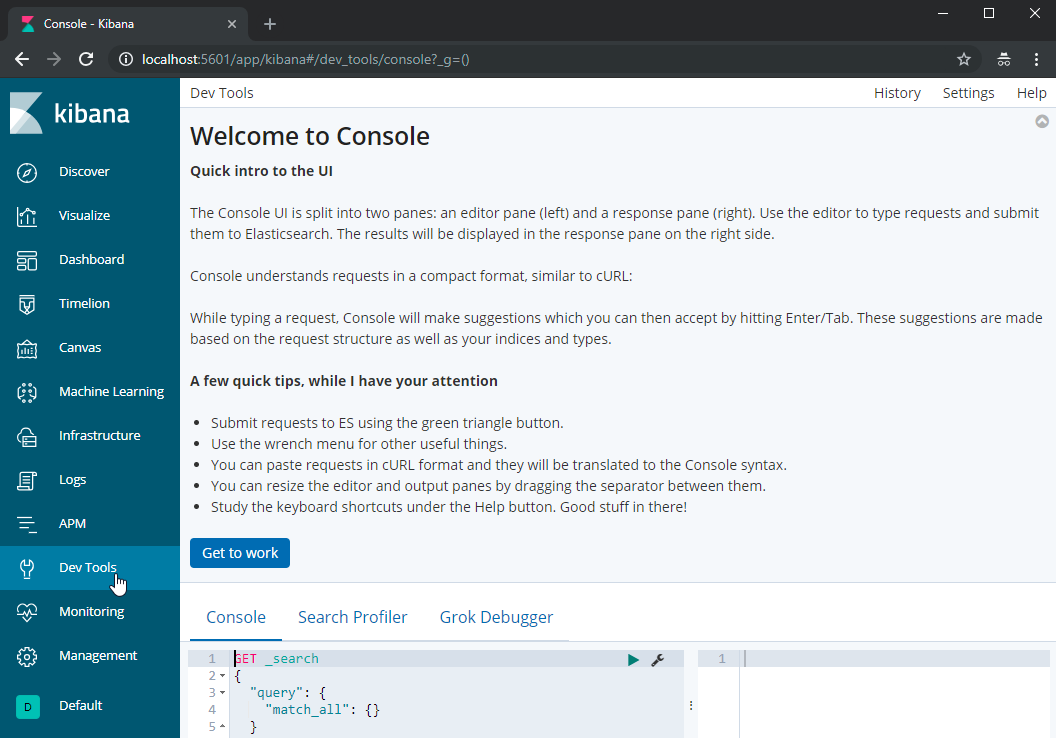
* « Titre » : titre du document
* « Contenu » : contenu du fichier PDF d’un document
* « Auteur » : auteur du document
* « MotsCles » : liste de mots-clés associés au document

Nous allons donc voir comment réaliser les opérations simples suivante :

* Création d’un document
* Modification d’un document
* Suppression d’un document
* Recherche de documents

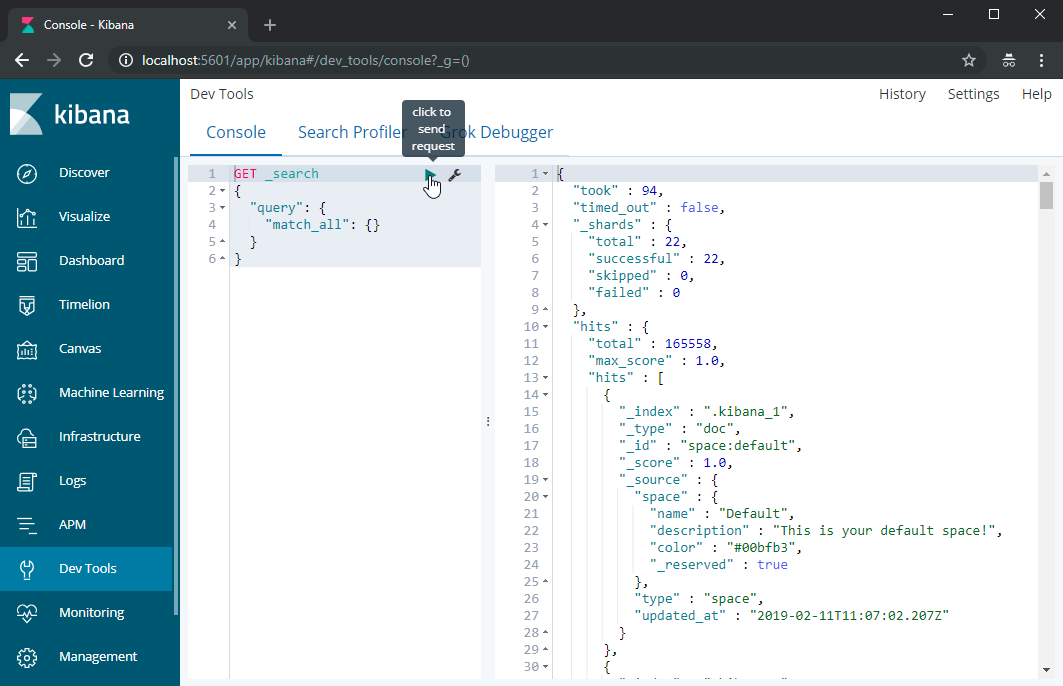
Warning : Pour davantage de détails sur la syntaxe à utiliser, je vous invite à vous référer à la [documentation officielle](<https://www.elastic.co/guide/en/elasticsearch/reference/current/docs.html>). Celle-ci présente clairement l’ensemble des opérations que je liste ci-dessous.

### ### Manipuler Elasticsearch avec la console DevTools

A travers cette 2ème étape, nous nous appuierons sur Kibana via son menu « DevTools » :

Une console est présentée permettant de simuler des interactions avec le moteur Elasticsearch :

* La partie gauche permet d’y inscrire sa requête à destination du moteur de recherche.
* La partie droite permet d’y vérifier la véracité du résultat.



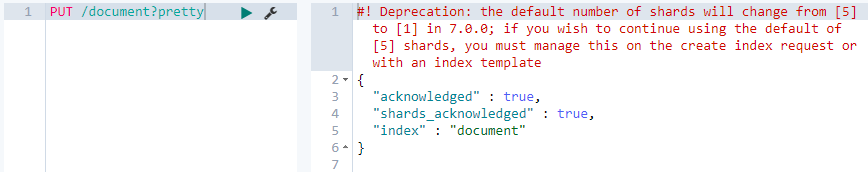
### ### Créer son premier document

#### #### Créer un index

Pour débuter, nous allons créer un index intitulé « document » avec la commande suivante :

|  |
| --- |
| PUT /document?pretty |

Cette commande crée l’index intitulé « document » comme le précise la sortie sous Kibana :



Warning : Il est possible de créer un index par l’intermédiaire de la commande suivante :

|  |
| --- |
| DELETE /document?pretty |

#### #### Ajouter un document

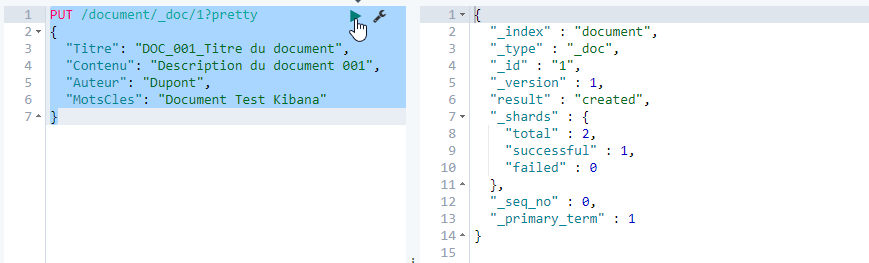
Maintenant que nous avons créé un index, nous allons maintenant insérer notre premier document. Nous y insérons les critères de recherche nécessaire :

* Le titre : « DOC\_001\_Titre du document »
* Le contenu : « Description du document 001 »
* L’auteur : « Dupont »
* Des mots clés : « Document Test Kibana »

Pour cela, utiliser la commande suivante :

|  |
| --- |
| PUT /document/\_doc/1?pretty  {  "Titre": "DOC\_001\_Titre du document",  "Contenu": "Description du document 001",  "Auteur": "Dupont",  "MotsCles": "Document Test Kibana"  } |

Cette commande insère le document comme le précise la sortie sous Kibana :



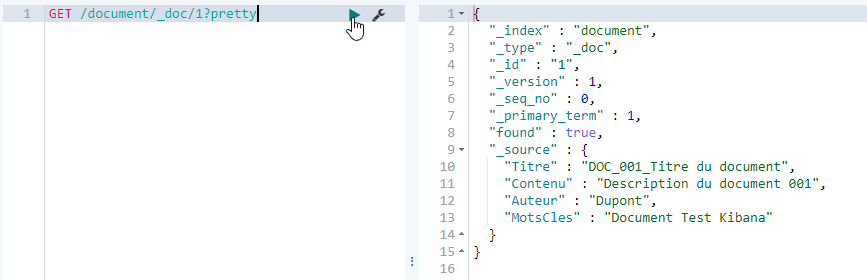
Warning : La valeur « successful » précise que la commande s’est réalisée avec succès et que celle-ci était une création via la mention « created » de la valeur « result ».

#### #### Récupérer un document

Maintenant que notre document est correctement inséré, nous pouvons le vérifier. Pour cela, utiliser la commande suivante :

|  |
| --- |
| GET /document/\_doc/1?pretty |

Cette commande insère le document comme le précise la sortie sous Kibana :



Warning : La valeur « \_source » précise le contenu du document.

### ### Modifier un document

Nous allons maintenant modifier l’une des valeurs d’un document grâce à la commande suivante :

|  |
| --- |
| POST /document/\_doc/1/\_update?pretty  {  "doc": { "Contenu": "Description du document 001 après une mise à jour" }  } |

Cette commande modifie le document comme le précise la sortie sous Kibana :



Warning : Le document est mis à jour et celui-ci incrémente sa version comme l’indique le champ « \_version ».

Vous pouvez vérifier sa bonne modification en vous référant à la commande de récupération d’un document.

### ### Rechercher un document

#### #### Ajouter des documents supplémentaires

La force d’Elasticsearch est bien dans ses capacités de requêtage. Nous allons effleurer ses capacités à partir de 2 types de recherche :

* Une recherche mono-critère classique
* Une recherche multi-critère basée sur des notions de pertinences

En amont de cette phase, je vous invite à y ajouter deux nouveaux documents. Pour ma part, je lance la commande suivante :

* Pour le 2nd document :

|  |
| --- |
| PUT /document/\_doc/2?pretty  {  "Titre": "DOC\_002\_Titre du document Word",  "Contenu": "Description du document 002 (au format autre que PDF)",  "Auteur": "Durant",  "MotsCles": "Document Test Word"  } |

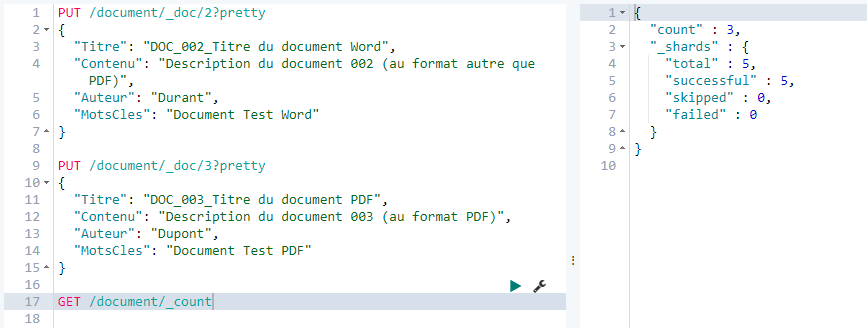
* Pour le 3ème document :

|  |
| --- |
| PUT /document/\_doc/3?pretty  {  "Titre": "DOC\_003\_Titre du document PDF",  "Contenu": "Description du document 003 (au format PDF)",  "Auteur": "Dupont",  "MotsCles": "Document Test PDF"  } |

Vous pouvez vérifier le bon remplissage de votre index via la commande suivante :

|  |
| --- |
| GET /document/\_count |

Cette commande vous précisera le nombre de documents via le champ « count » :



Nous disposons donc des données suivantes au sein du moteur Elasticsearch :

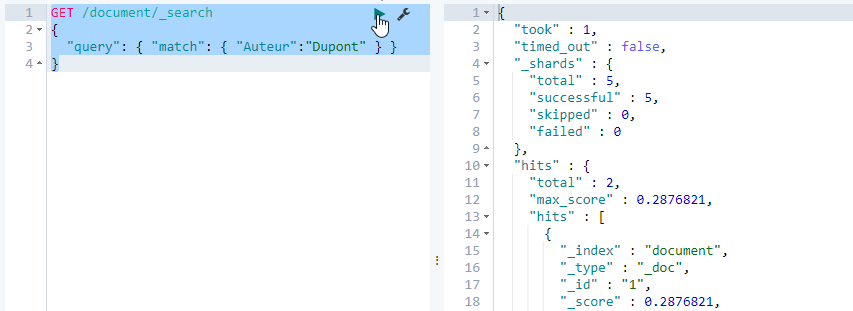
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Titre | Contenu | Auteur | MotsCles |
| 1er document | DOC\_001\_Titre du document | Description du document 001 après une mise à jour | Dupont | Document Test Kibana |
| 2ème document | DOC\_002\_Titre du document Word | Description du document 002 (au format autre que PDF) | Durant | Document Test Word |
| 3ème document | DOC\_003\_Titre du document PDF | Description du document 003 (au format PDF) | Dupont | Document Test PDF |

#### #### Effectuer une recherche mono-critère

Nous allons rechercher la liste des documents dont l’auteur est « Dupont » via la commande suivante :

|  |
| --- |
| GET /document/\_search  {  "query": { "match": { "Auteur":"Dupont" } }  } |

Cette commande liste l’ensemble des documents en accord avec le critère précisé :



Suite à la recherche, Elasticsearch donne les informations suivantes :

* « hits » : la liste des documents
* « total » : le nombre de documents validant les critères demandés
* « max\_score » : le score de pertinence de la donnée jugée la plus pertinente avec la recherche demandée
* « \_score » : pour chaque document retourné, il précise la pertinence de la donnée au regard de la recherche demandée

Au vu de la recherche demandée, les 2 documents disposent du même degré de pertinence.

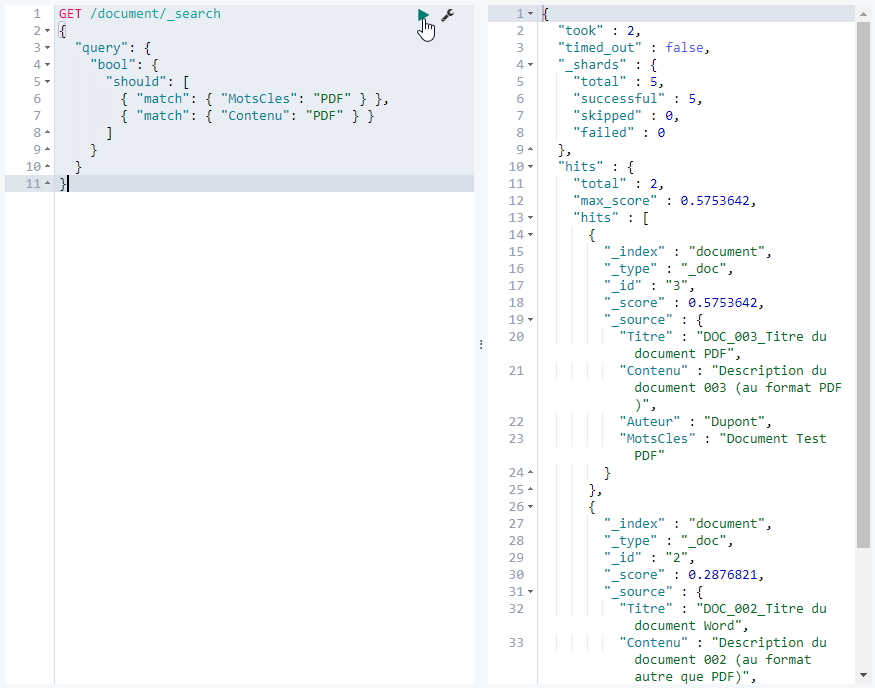
#### #### Effectuer une recherche mono-critère

Nous allons désormais rechercher la liste des documents mentionnant la notion « PDF ». Nous voulons identifier le document PDF le plus approprié à ma recherche.

|  |
| --- |
| GET /document/\_search  {  "query": {  "bool": {  "should": [  { "match": { "MotsCles": "PDF" } },  { "match": { "Contenu": "PDF" } }  ]  }  }  } |

Cette commande recherche les documents dont les mots clés et le contenu contiennent le terme « PDF ». Via l’instruction « should », cette recherche dispose d’une tolérance dans les résultats qu’elle propose.

Cette commande liste l’ensemble des documents en accord avec le critère précisé :



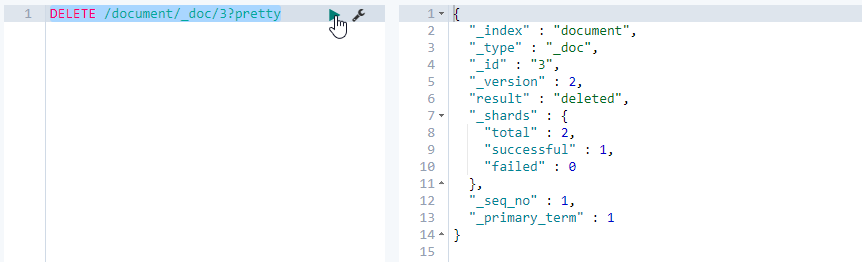
Suite à la recherche, on peut voir que cette recherche donne 2 résultats. Ces résultats disposent d’un score de pertinence qui diverge. L’un des documents dispose d’un score plus élevé que le second.

### ### Supprimer un document

Nous allons maintenant supprimer un document grâce à la commande suivante :

|  |
| --- |
| DELETE /document/\_doc/3?pretty |

Cette commande supprime le document comme le précise la sortie sous Kibana :



Warning : Le document est supprimé et l’index « document » est désormais constitué de 2 tuples.

## ## Conclusion

A travers ce billet, j’espère vous avoir permis de mettre un pied à l’étrier dans l’utilisation d’Elasticsearch. Grâce à ces quelques lignes, vous avez ainsi été en mesure de mettre en place puis de tester Elasticsearch directement via Kibana.

Dans mon cadre d’utilisation, mon but était simplement de prendre en main l’outil et de comprendre comment celui-ci fonctionnait. Je n’ai donc réalisé que des manipulations simples sur Elasticsearch..

Je vous invite à consulter la documentation officielle pour aller plus loin. Je l’ai jugé assez claire et bien structurée.