

TP1 ElasticSearch

L'installation d'ElasticSearch est assez simple :

- Récupérer ElasticSearch depuis le partage réseau
- Dézipper
- Lancer bin/elasticsearch.bat (sous windows) ou bin/elasticsearch (sous linux)

Si par malheur vous avez l'erreur **Unsupported major.minor version 51.0**, vérifiez bien que vous possédez une des dernières version de Java 7 ou Java 8. Vérifiez aussi que la variable d'environnement JAVA_HOME pointe sur une de ces versions.

Pour tester qu'ElasticSearch est bien up, il suffit de taper dans un navigateur : <http://localhost:9200/> Et d'avoir la réponse :

```
{
  name: "Batragon",
  cluster_name: "elasticsearch",
  cluster_uuid: "eR8nfkLUR0eeR9zjGOJ_Yw",
  version: {
    number: "2.4.4",
    build_hash: "fcbb46dfd45562a9cf00c604b30849a6dec6b017",
    build_timestamp: "2017-01-03T11:33:16Z",
    build_snapshot: false,
    lucene_version: "5.5.2"
  },
  tagline: "You Know, for Search"
}
```

Consoles d'administration

Avant d'aller plus loin, nous allons installer des consoles d'administration pour suivre ce que fait ElasticSearch. Il existe plusieurs plugins « console » et j'en ai choisi 2 (à vous de tester pour voir celui qui vous convient le mieux)

elastiqHQ

Aller dans le répertoire d'installation d'ElasticSearch et exécuter

```
bin/plugin.bat install royrusso/elasticsearch-HQ (sous windows)
```

```
bin/plugin -install royrusso/elasticsearch-HQ (sous linux)
```

La console sera accessible à l'adresse : http://localhost:9200/_plugin/HQ/ (une fois ElasticSearch démarré)

elasticsearch-head

Aller dans le répertoire d'installation d'ElasticSearch et exécuter

```
bin/plugin.bat install mobz/elasticsearch-head (sous windows)
```

```
bin/plugin -install mobz/elasticsearch-head (sous linux)
```

La console sera accessible à l'adresse : http://localhost:9200/_plugin/head/ (une fois ElasticSearch démarré)

bigDesk

Aller dans le répertoire d'installation d'ElasticSearch et exécuter

```
bin/plugin.bat install nishantsaini/bigdesk (sous windows)
```

```
bin/plugin -install nishantsaini/bigdesk (sous linux)
```

La console sera accessible à l'adresse : http://localhost:9200/_plugin/bigdesk/ (une fois ElasticSearch démarré)

Cette console est particulièrement utile pour configurer les clusters/nœuds/shards ...

Histoires de nœuds et de cluster

Maintenant que nous sommes un minimum outillés nous pouvons parler de l'architecture d'ElasticSearch et la « voir » !

ElasticSearch est un moteur d'indexation open source basé sur Lucene. Il permet d'indexer des données sous forme de documents, contrairement à Solr (autre moteur de recherche basé sur Lucene) qui gère des structures « à plat ».

Un fichier CSV a une structure à plat :

```
nom;code postal;url
```

```
paris;75001;www.paris.fr
```

```
boulogne-billancourt;92100 ;http://www.boulognebillancourt.com/
```

Les fichiers XML ou JSON ont une structure documentaire :

```
<ville>
  <nom>paris</nom>
  <codePostal>75001</codePostal>
  <url>http://www.paris.fr</url>
</ville>
<ville>
  <nom>boulogne-billancourt</nom>
  <codePostal>92100</codePostal>
  <url>http://www.boulognebillancourt.com/</url>
</ville>
```

ElasticSearch intègre une API REST au format JSON qui permet de réaliser les fonctionnalités d'indexation, de recherche et de configuration.

Lorsqu'on démarre une instance d'ElasticSearch comme nous l'avons fait plus haut, on démarre en fait un nœud (**node**). Ce nœud fait partie d'un **cluster**.

Un nœud est partitionné en **shards**. Quand on demande l'indexation d'un document il est stocké dans un shard.

Un shard est un indexe Lucene à part entière qui a besoin de ressources pour pouvoir fonctionner correctement (RAM, Processeur, ...).

Un indexe ElasticSearch est donc constitué de plusieurs indexes Lucene, ce qui permet de répartir les documents.

Lorsqu'un document est indexé, il est tout d'abord stocké dans un seul shard primaire.

Une fois que le document est indexé dans un shard primaire, il est répliqué dans 0 ou plusieurs shards replica. Mais pour illustrer tout ça nous allons suivre quelques étapes :

1- Premier démarrage d'ElasticSearch

Au premier démarrage d'une instance d'ElasticSearch, un noeud est créé avec un nom aléatoirement choisi parmi 3000 noms des personnages Marvel.

Dans mon exemple « Batragon ». Ce nœud cherche sur le réseau s'il existe un cluster appelé « elasticsearch » (original). Ce n'est pas le cas, alors il crée un cluster et se déclare master à l'intérieur de ce cluster.

Par défaut le nœud est partitionné en 5 shards. Ci-dessous les logs :

En illustrations ça donne :

The screenshot displays the Elasticsearch Kibana dashboard. At the top, the URL is 'http://localhost:9200/'. The main header shows 'elasticsearch' with a green status bar indicating 'Santé du cluster: green (0.0)'. Below this, there are tabs for 'Aperçu', 'Index', 'Navigateur', 'Recherche Structurée [+]', and 'Autres requêtes [+]', along with an 'Info' dropdown. The 'Aperçu du cluster' section is active, showing a 'Sort Cluster' dropdown, 'View Aliases', and an 'Index Filter' input. A 'Rafraîchir' button is also present. The node information section shows a star icon, the node name 'Dmitri Smerdyakov', and buttons for 'Info' and 'Actions'. A red label 'Nom du noeud' points to the node name. Below this, a red label 'Indication du Master' points to the node name. The 'Illustration plugin elasticsearch-head' section shows the 'Elastic HQ' logo, a search bar with 'http://localhost:9200' and a 'Connect' button, and links to 'Star us on GitHub' and 'Blog'. The bottom section shows the 'elasticsearch' logo, a 'Node Diagnostics' button, and a 'Dmitri Smerdyakov' button. A red label 'Nom du cluster' points to the 'elasticsearch' logo.

Illustration plugin elasticsearch-head

Illustration plugin Elastic HQ

2- Modification du cluster et du noeud

Nous allons faire une première modification dans le fichier de configuration d'ElasticSearch.

Aller dans conf/**elasticsearch.yml** et mettre

```
cluster.name: cluster-blog  
node.name: "blog-01"
```

Redémarrer ElasticSearch et constater la modification dans une des consoles.

3- Indexation d'un document

Comme mentionné ci-dessus nous pouvons indexer un document via un service REST JSON.

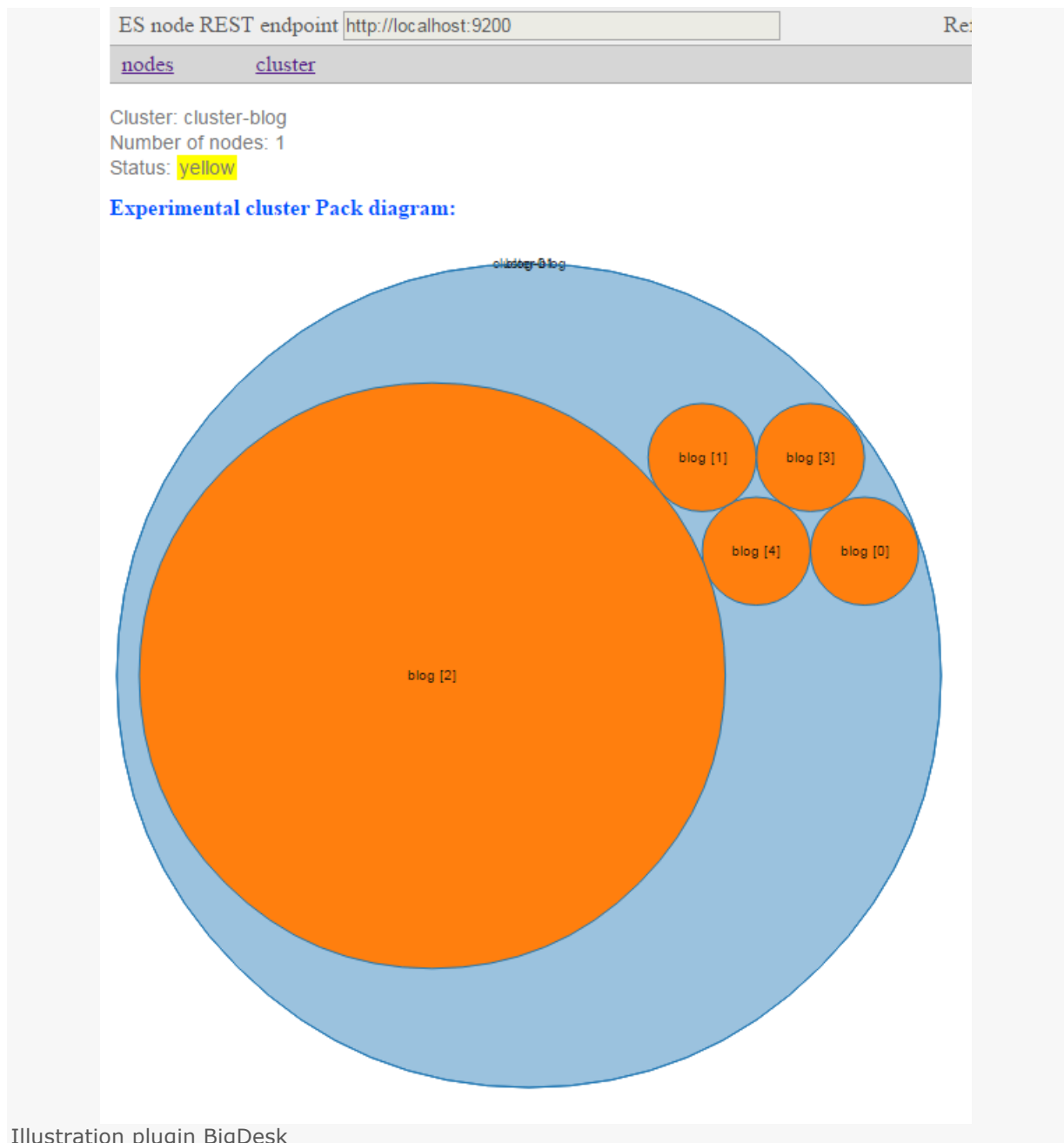
Par exemple pour créer le document avec l'id 1, de type city1 dans l'index blog, il suffit d'exécuter la commande :

```
curl -XPUT 'http://localhost:9200/blog/city1/1' -d '{  
  "nom": "Paris",  
  "codePostal": "75000",  
  "url": "http://www.paris.fr/",  
  "monuments": [  
    {  
      "nom": "Arc de Triomphe"  
    },  
    {  
      "nom": "Tour Eiffel"  
    }  
  ]  
}'
```

Par défaut ElasticSearch indexe tous les champs. Regardons nos consoles :

The screenshot shows the Kibana interface for an ElasticSearch cluster named 'cluster-blog'. The cluster health is 'Santé du cluster: yellow (5 10)'. The 'blog' index is selected, showing a size of 3.74ki and 1 document. The 'Aperçu du cluster' section shows two nodes: 'Unassigned' and 'blog-01'. The 'blog-01' node is highlighted with a star and shows a green status for all five shards (0, 1, 2, 3, 4).

Illustration après indexation d'un document



Dans notre nœud blog-01, 5 shards ont été créés : 0, 1, 2, 3, 4, 5.

Dans ElasticSearch Head, on peut cliquer sur un shard pour avoir le détail (on voit qu'il y a 1 document indexé). On remarque dans la console une indication sur la santé du cluster :

- **Rouge** : On ne peut ni indexer, ni rechercher. Le système est HS.
- **Jaune** : On peut indexer et rechercher, par contre si l'instance/nœud ElasticSearch s'arrête, plus rien ne fonctionne.
- **Vert** : On peut indexer et rechercher. Si l'instance/nœud ElasticSearch s'arrête ce n'est pas grave car il existe un ou plusieurs autres nœuds où les données ont été répliquées.

4- Ajout d'un nœud

Justement pour passer notre indicateur au vert et surtout rendre notre installation plus robuste nous allons ajouter un nœud dans notre cluster.

Pour les tests on va juste copier/coller le dossier d'installation d'ElasticSearch dans un dossier bis. En environnement de recette/production on déploierait une nouvelle instance sur un autre serveur du même réseau. Dans l'installation bis, il suffit d'éditer le fichier de configuration elasticsearch.yml et de mettre :

```
node.name: "blog-02"
```

Il ne faut surtout pas changer le nom du cluster. On supprime aussi les dossiers data et logs de cette installation bis et on la démarre !

Ce nouveau nœud va chercher s'il existe sur le réseau un cluster avec le même nom : « cluster-blog ». Comme il existe, il va le rejoindre.

Logs du nœud blog-02

```
[2014-12-12 00:41:51,892][INFO ][node ] [blog-02] initialized ...
```

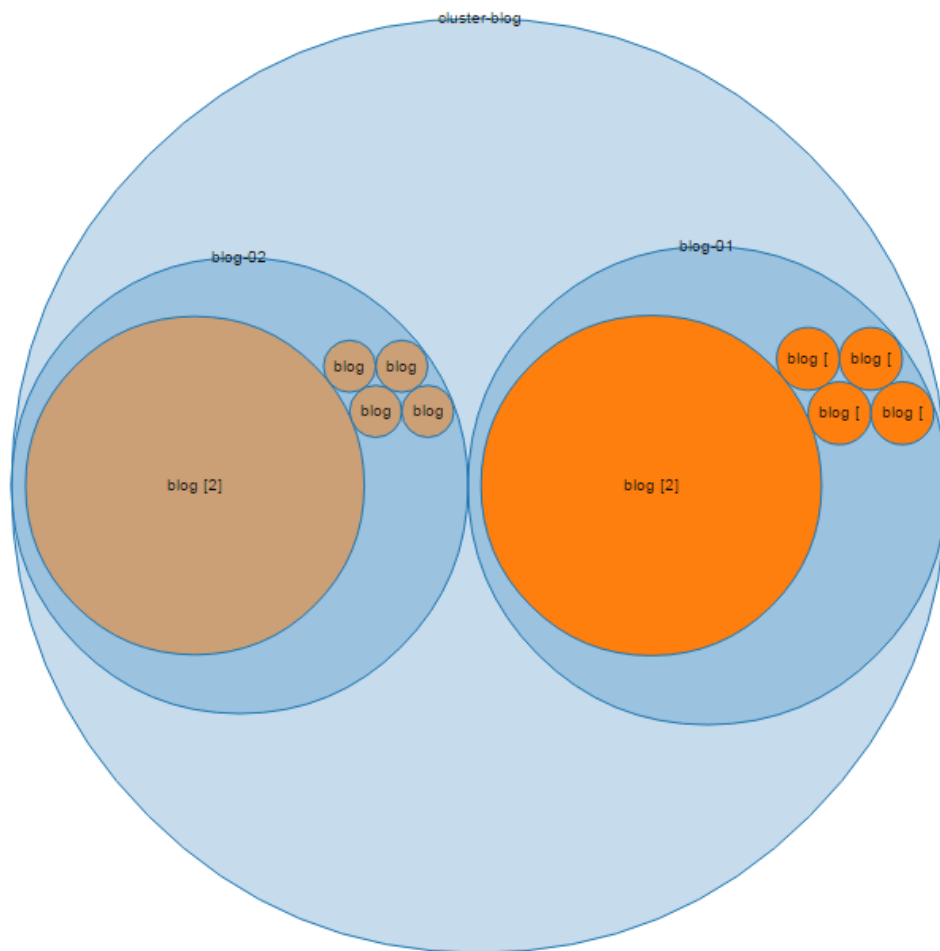
```
[2014-12-12 00:41:52,433][INFO ][discovery ] [blog-02] cluster-blog/4NP2RtaORHuUSttFIEJgcg
[2014-12-12 00:41:55,541][INFO ][cluster.service ] [blog-02] detected_master [blog-01]
[Yd6yUrUCRoeSYFV8GUOexQ][globedevoreur][inet[/192.168.1.5:9300]], added {[blog-01]
[Yd6yUrUCRoeSYFV8GUOexQ][globedevoreur][inet[/192.168.1.5:9300]],}, reason: zen-disco-
receive(from master [[blog-01][Yd6yUrYFV8GUOexQ][globedevoreur][inet[/192.168.1.5:9300]]])
```

Logs du nœud blog-01

```
[2014-12-12 00:41:55,503][INFO ][cluster.service ] [blog-01] added {[blog-02]
[4NP2RtaORHuUSttFIEJgcg][globedevoreur][inet[/192.168.1.5:9301]] ,}, reason: zen-disco-
receive(join from node[[blog-02][4NP2RtaORHuUSttFIEJgcg][globedevoreur][inet[/192.168.1.5:9301]]])
```

Et en image cela donne :

The screenshot shows the ElasticSearch Head web interface. At the top, the URL is 'http://localhost:9200/'. The cluster name is 'cluster-blog' and the health status is 'Santé du cluster: green (10/10)'. Below the navigation bar, there's a section for 'Aperçu du cluster' with buttons for 'Sort Cluster' and 'View Aliases'. The main content area shows a list of nodes. The first node is 'blog-01' with a star icon, and the second is 'blog-02' with a circle icon. Each node has a status bar with five colored squares (0, 1, 2, 3, 4) representing different components. Below the nodes, it says 'ElasticSearchHead avec 2 nœuds'.



BigDesk avec 2 nœuds

Remarques :

- L'indicateur est au vert... c'est bon signe !
- Le nœud blog-02 a bien rejoint le cluster cluster-blog
- BigDesk nous montre que le shard numéro 2 du deuxième nœud s'est rempli automatiquement : le document que nous avons indexé a été répliqué dans ce shard, aussi appelé **shard replica**. Il est possible de configurer plusieurs shards replica pour un seul shard primaire (peut améliorer le temps de réponse si le système est très sollicité en recherches).

5- Faire une recherche

Ci-dessous un exemple pour faire une recherche du mot « eiffel » dans l'index en utilisant l'API REST:

```
curl -XGET 'http://localhost:9200/blog/city1/_search?q=eiffel'
```

Et la réponse où on retrouve bien notre document :

```
{ "took":147,"timed_out":false,"_shards":{"total":5,"successful":5,"failed":0},
  "hits":{"total":1,"max_score":0.095891505,"hits":
    [{ "_index":"blog", "_type":"city1", "_id":"1", "_score":0.095891505, "_source":{
      "nom": "Paris",
      "codePostal":"75000",
      "url": "http://www.paris.fr/",
      "monuments": [
        {
          "nom": "Arc de Triomphe"
        },
        {
          "nom": "Tour Eiffel"
        }
      ]
    }
  ]
}}
```

6- Arrêter un nœud

Pour finir sur l'histoire des nœuds et cluster, nous pouvons arrêter le nœud blog-01 (en stoppant la première instance d'ElasticSearch).

Dans les logs du nœud blog-02, on voit que celui-ci devient master automatiquement (c'est la configuration par défaut).

L'indexation et la recherche sont donc encore possibles !

```
[2014-12-12 00:50:41,083][WARN ][discovery.zen ] [blog-02] master left (reason =
shut_down), current nodes: {[blog-02][4NP2RtaORHuUSttFIEJgcg][globedevoreur][inet[/192.168.1.5:9301]],}
```

```
[2014-12-12 00:50:41,088][INFO ][cluster.service ] [blog-02] removed {[blog-
01][Yd6yUrUCRoeSYFV8GUOexQ][globedevoreur][inet[/192.168.1.5:9300 ]],}, reason: zen-disco-
master_failed ([blog-01][Yd6yUrUCRoeSYFV8GUOexQ][globedevoreur][inet[/192.168.1.5:9300]])
```

```
[2014-12-12 00:50:44,864][INFO ][cluster.service ] [blog-02] new_master [blog-
02][4NP2RtaORHuUSttFIEJgcg][globedevoreur][inet[/192.168.1.5:9301]], reason: zen-disco-join
(elected_as_master)
```

Conclusion

Dans cette première partie, nous avons décortiqué pas à pas l'architecture d'ElasticSearch. Nous avons pu voir les avantages suivants :

- Installation très facile (exemple fait sur Windows, mais fonctionne tout aussi bien sur Linux ou Mac Os)
- Outil très « scalable ». L'ajout d'un nœud se fait très rapidement et permet d'avoir des données répliquées et accessibles même si certains serveurs s'arrêtent.
- Des plugins sympas qui sont un gros plus pour manipuler ElasticSearch

Il reste à « brancher » ElasticSearch sur MongoDB et l'utiliser dans de vraies conditions.