



elasticsearch

<http://www.elasticsearch.org/>

(Near) Real Time Distributed Search Engine

David Pilato - @dadoonet / @ElasticsearchFR

Qui suis-je ?

- Ecole Nationale d'Ingénieurs de Brest – Promo 1994
- 3 ans en SSII, bon à tout faire
- 3 ans chez Cegetel/SFR, touche à tout
- 4 ans chez e-Brands, startup internet des années 2000, chef de projets
- Depuis 2005, DGDDI, mouton à 5 pattes
 - Projet de déclaration de douane en ligne (entreprises)
 - Projet de gestion des contrôles de la douane (douaniers)



Pourquoi un moteur ?

- Un document dans une base de données
 - Un attribut date :
 - 15/12/2011
 - Un attribut codifié Pays :
 - FR
 - Correspondant à la table d'association code/libellé
 - Code : FR
 - Libellé : France
 - Un attribut commentaire :
 - J'observe une erreur de saisie dans la désignation commerciale du produit. Songer à téléphoner à David.



Pourquoi un moteur ?

- Cherche moi un document de décembre 2011 portant sur la France et contenant saisie et David



Pourquoi un moteur ?

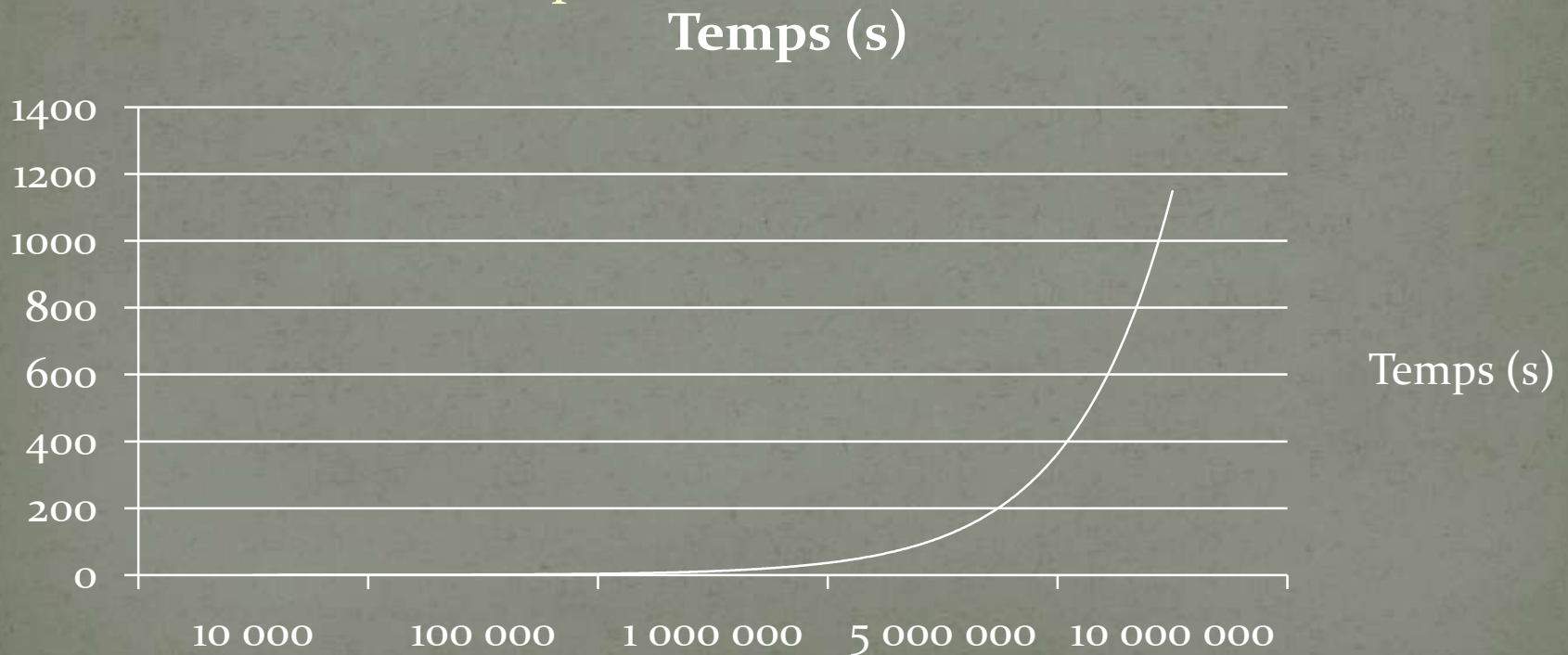
- Cherche moi un document de décembre 2011 portant sur la France et contenant saisie et David
 - En SQL

```
SELECT
  doc.*, pays.*
FROM
  doc, pays
WHERE
  doc.pays_code = pays.code AND
  doc.date_doc > to_date('2011-12', 'yyyy-mm') AND
  doc.date_doc < to_date('2012-01', 'yyyy-mm') AND
  pays.libelle = 'France' AND
  doc.commentaire LIKE '%saisie%David%';
```



Pourquoi un moteur ?

- Cherche moi un document de décembre 2011 portant sur la France et contenant saisie et David
 - Performances espérées



Pourquoi un moteur ?

- Cherche moi un document de décembre 2011 portant sur la france et contenant saisie ou david



Pourquoi un moteur ?

- Cherche moi un document de décembre 2011 portant sur la france et contenant saisie ou david
 - En SQL

```
SELECT
  doc.*, pays.*
FROM
  doc, pays
WHERE
  doc.pays_code = pays.code AND
  doc.date_doc > to_date('2011-12', 'yyyy-mm') AND
  doc.date_doc < to_date('2012-01', 'yyyy-mm') AND
  lower(pays.libelle) = lower('France') AND (
    lower(doc.commentaire) LIKE lower('%saisie%') OR
    lower(doc.commentaire) LIKE lower('%David%'));
```



Pourquoi un moteur ?

- Plus fun : cherche moi un document qui idéalement sera de décembre 2011, aura les mots France, saisie ou david dans quelque part, et indique moi d'abord les éléments les plus pertinents



Pourquoi un moteur ?

- Plus fun : cherche moi un document qui idéalement sera de décembre 2011, aura les mots France, saisie ou david dans quelque part, et indique moi d'abord les éléments les plus pertinents
 - En SQL :

```
SELECT
  doc.*, pays.*
FROM
  doc, pays
WHERE
  doc.pays_code = pays.code AND
  doc.date_doc > to_date('2011-12', 'yyyy-mm') AND
  doc.date_doc < to_date('2012-01', 'yyyy-mm') AND
  lower(pays.libelle) = lower('France') AND (
    lower(doc.commentaire) LIKE lower('%saisie%') OR
    lower(doc.commentaire) LIKE lower('%david%'));

SELECT
  doc.*, pays.*
FROM
  doc, pays
WHERE
  doc.pays_code = pays.code AND
  doc.date_doc > to_date('2011-12', 'yyyy-mm') AND
  doc.date_doc < to_date('2012-01', 'yyyy-mm') AND
  lower(pays.libelle) = lower('France') AND (
    lower(doc.commentaire) LIKE lower('%saisie%') OR
    lower(doc.commentaire) LIKE lower('%david%'));

SELECT
  doc.*, pays.*
FROM
  doc, pays
WHERE
  doc.pays_code = pays.code AND
  doc.date_doc > to_date('2011-12', 'yyyy-mm') AND
  doc.date_doc < to_date('2012-01', 'yyyy-mm') AND
  lower(pays.libelle) = lower('France') AND (
    lower(doc.commentaire) LIKE lower('%saisie%') OR
    lower(doc.commentaire) LIKE lower('%david%'));
```

MISSION IMPOSSIBLE



Un moteur de recherche ?

- Avant tout, un moteur d'indexation de documents et un moteur de recherche sur les index

TODO



Key features

- Basé sur Lucene
- HTTP / REST
- JSON
- Utilisable à partir de n'importe quelle technologie
- Distribué
- Partitionnement
- Multi index
- Taillé pour le cloud



Quelques notions

- Index : Espace de stockage des documents
- Document : Un objet représentant les données (au sens NoSQL)
- Mapping : représentation des attributs d'un document avec leurs propriétés d'indexation
- Partition (shard) : permet de découper un index en plusieurs parties pour y distribuer les documents
- Réplication (replica) : recopie d'une partition en une ou plusieurs copies



Installons ElasticSearch

Download ES at <http://www.elasticsearch.org/download>

```
$ curl -o elasticsearch.zip
```

```
  http://cloud.github.com/downloads/elasticsearch/elasticsearch/elasticsearch-0.17.8.zip
```

Unzip it

```
$ unzip elasticsearch.zip
```

We create two instances on the same machine

```
$ cp -r elasticsearch-0.17.8 es1
```

```
$ mv elasticsearch-0.17.8 es2
```

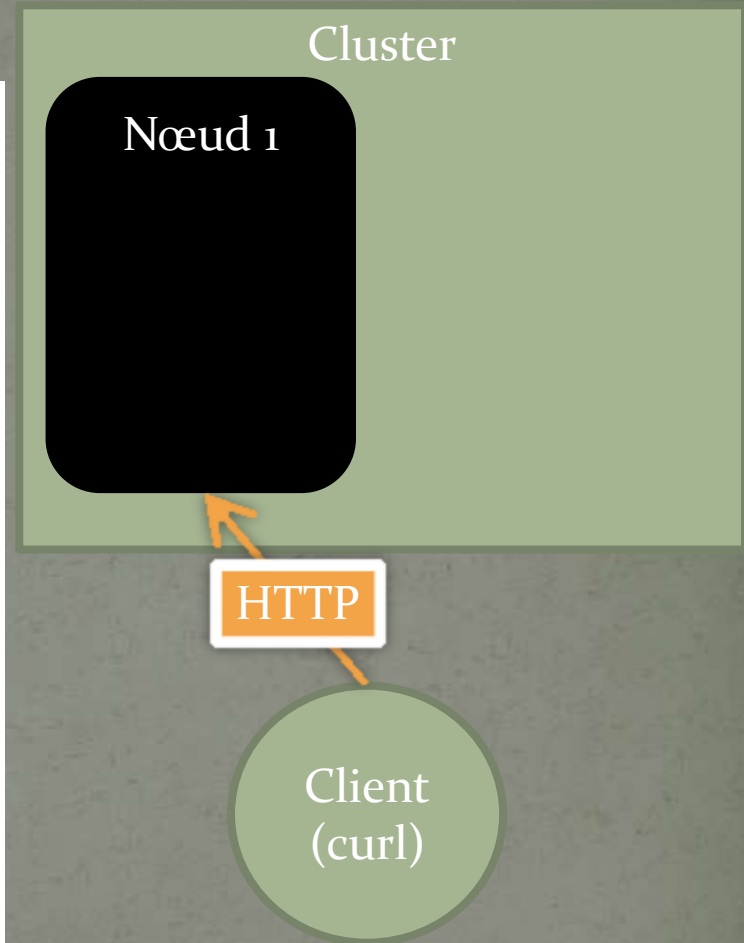
We start in foreground (for the demo) the first instance

```
$ es1/bin/elasticsearch -f
```



Client Elasticsearch

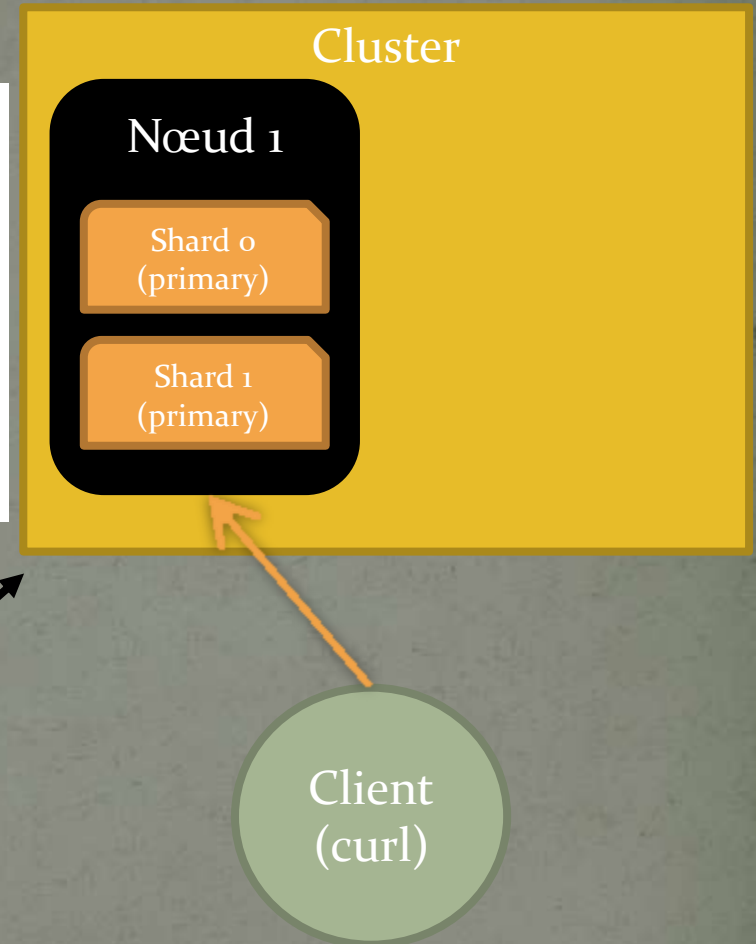
```
$ curl localhost:9200
```



Créons un index

```
$ curl -XPUT localhost:9200/parisjug -d '{  
  "index" : {  
    "number_of_shards" : 2,  
    "number_of_replicas" : 1  
  }  
'
```

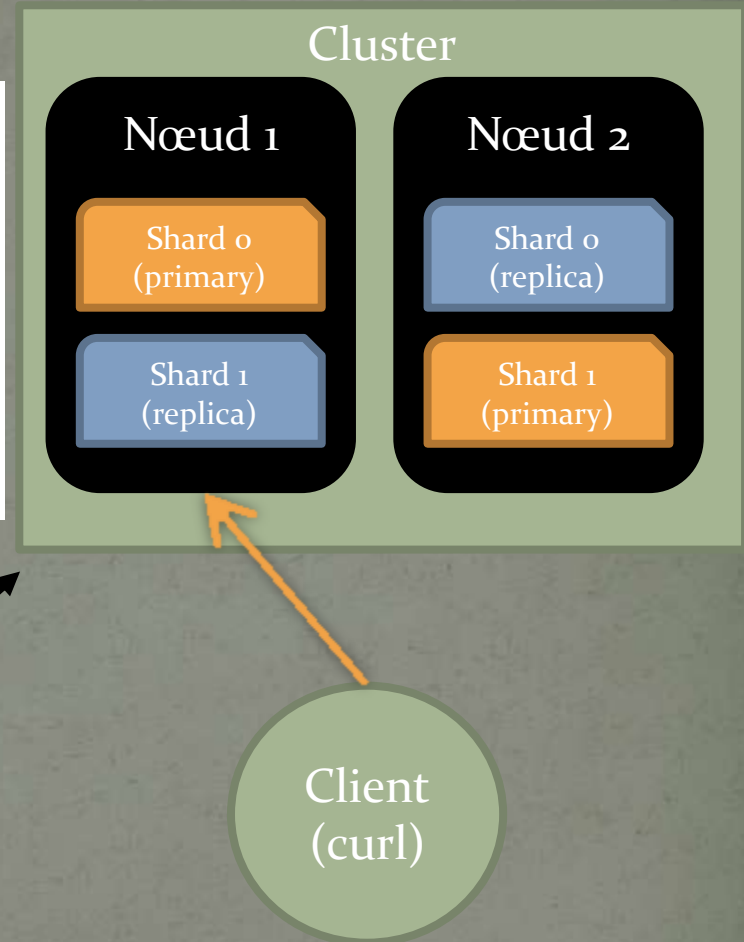
Cluster passe en **YELLOW**
réplication non respectée



Ajoutons un noeud

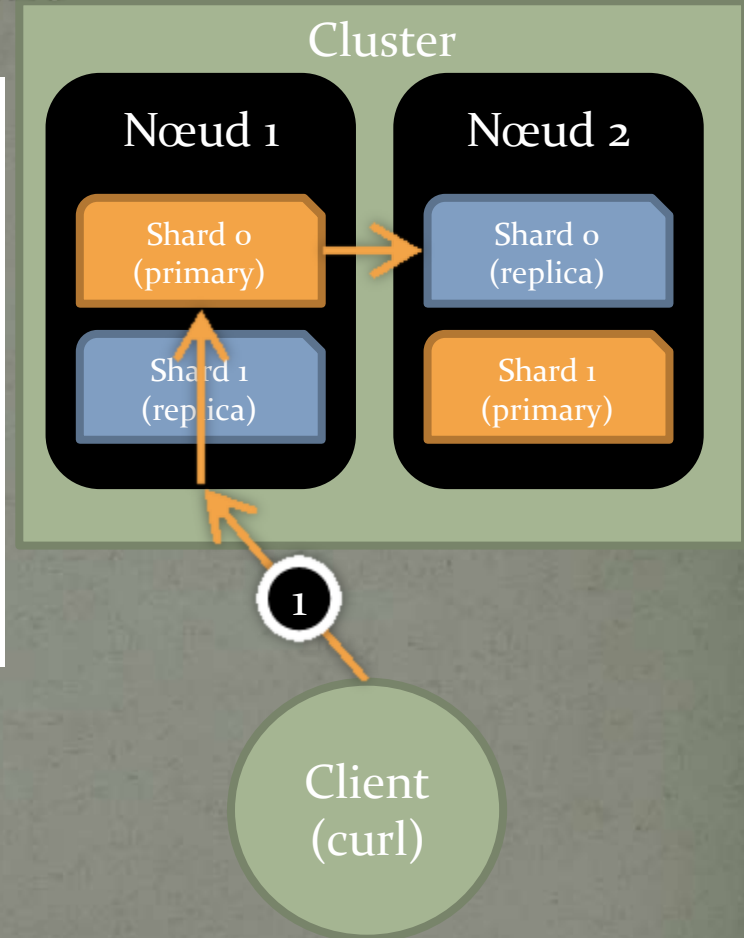
```
$ es2/bin/elasticsearch -f
```

Cluster passe en **GREEN**
réplication respectée



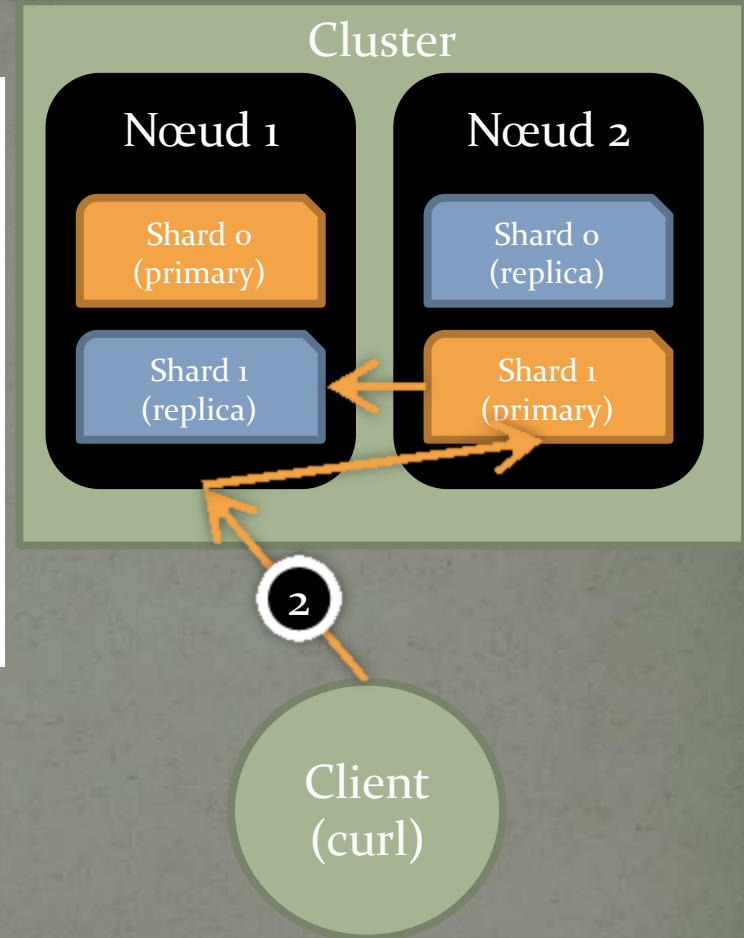
Indexons un document

```
$ curl -XPUT localhost:9200/parisjug/talk/1 -d '{
  "name" : {
    "first" : "Emmanuel",
    "last" : "Bernard"
  } ,
  "title" : "Hibernate Search"
}'
```



Indexons un document

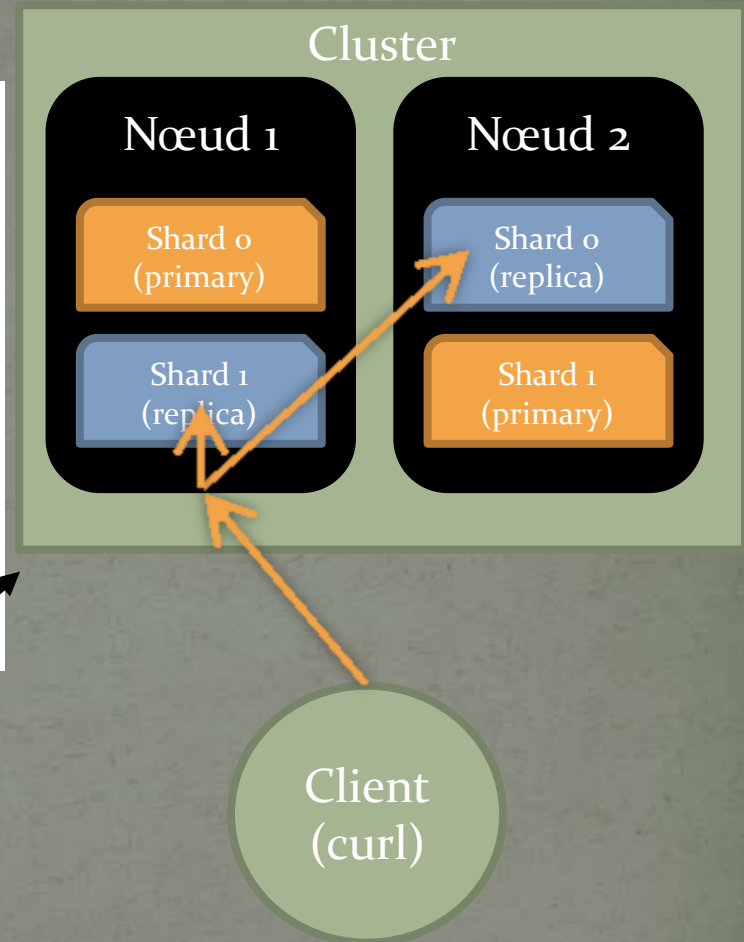
```
$ curl -XPUT localhost:9200/parisjug/talk/2 -d '{
  "name" : {
    "first" : "David",
    "last" : "Pilato"
  } ,
  "title" : "Elastic Search"
}'
```



Cherchons !

```
$ curl localhost:9200/parisjug/_search?q=search
```

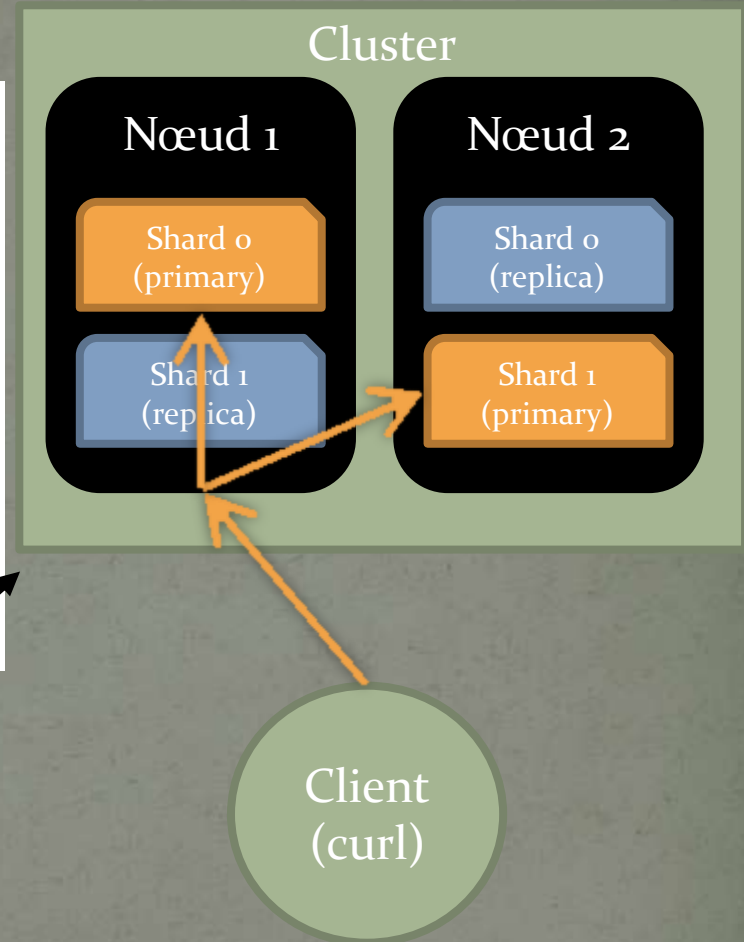
La recherche est distribuée



Cherchons encore !

```
$ curl localhost:9200/parisjug/_search?q=search
```

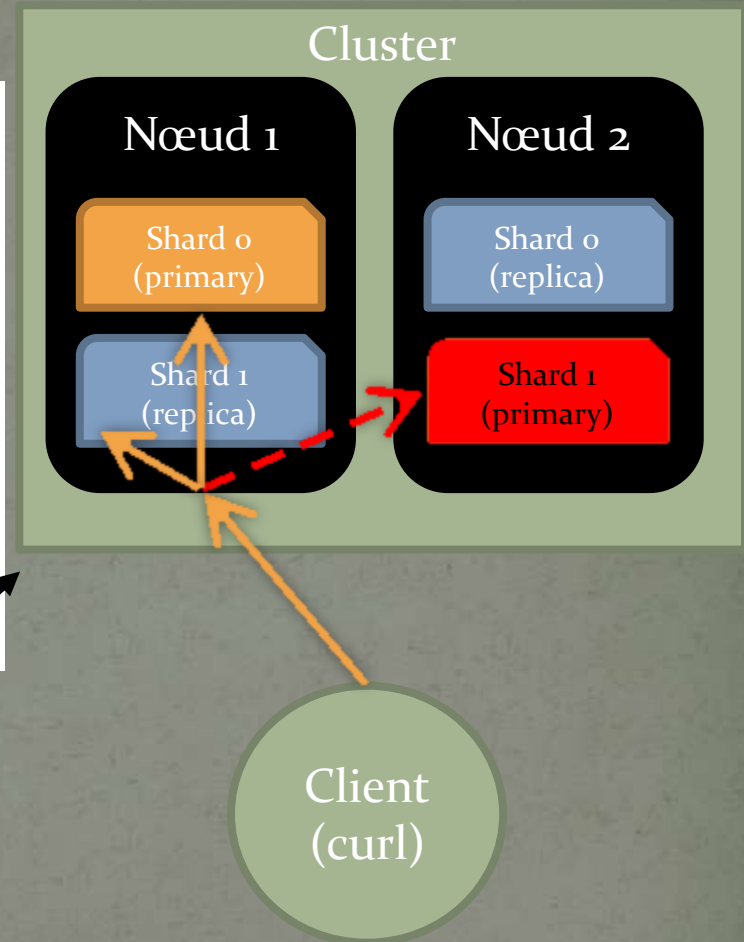
Automatic balancing



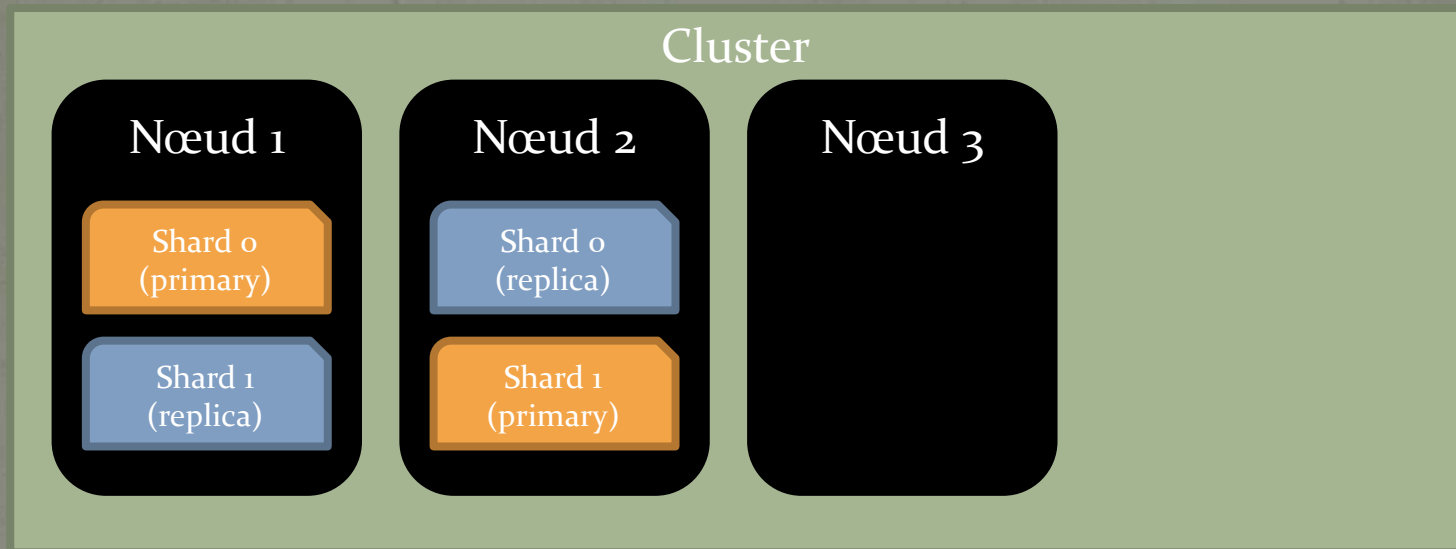
Cherchons toujours !

```
$ curl localhost:9200/parisjug/_search?q=search
```

Automatic failover



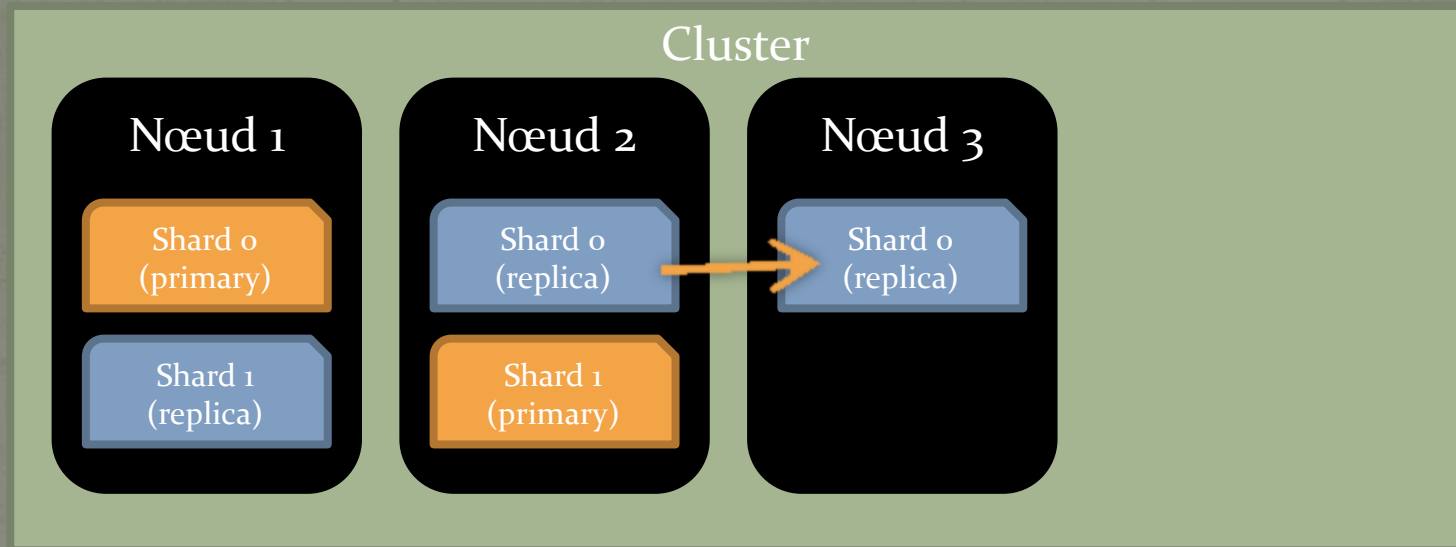
Ajoutons un nœud



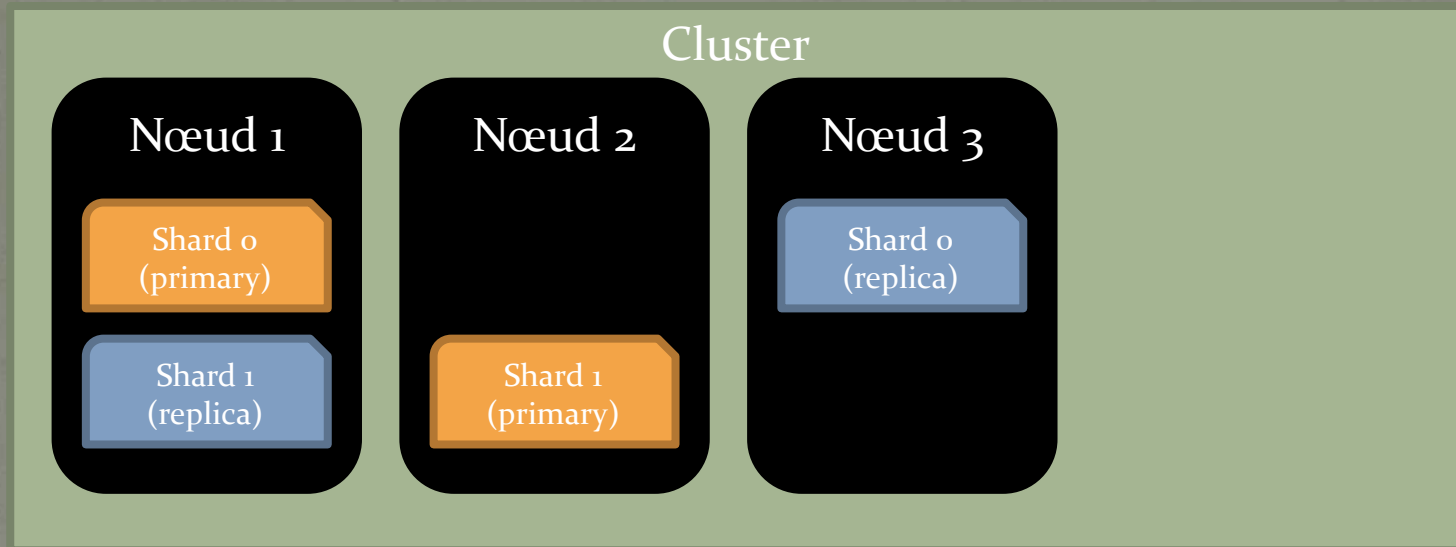
Réallocation dynamique



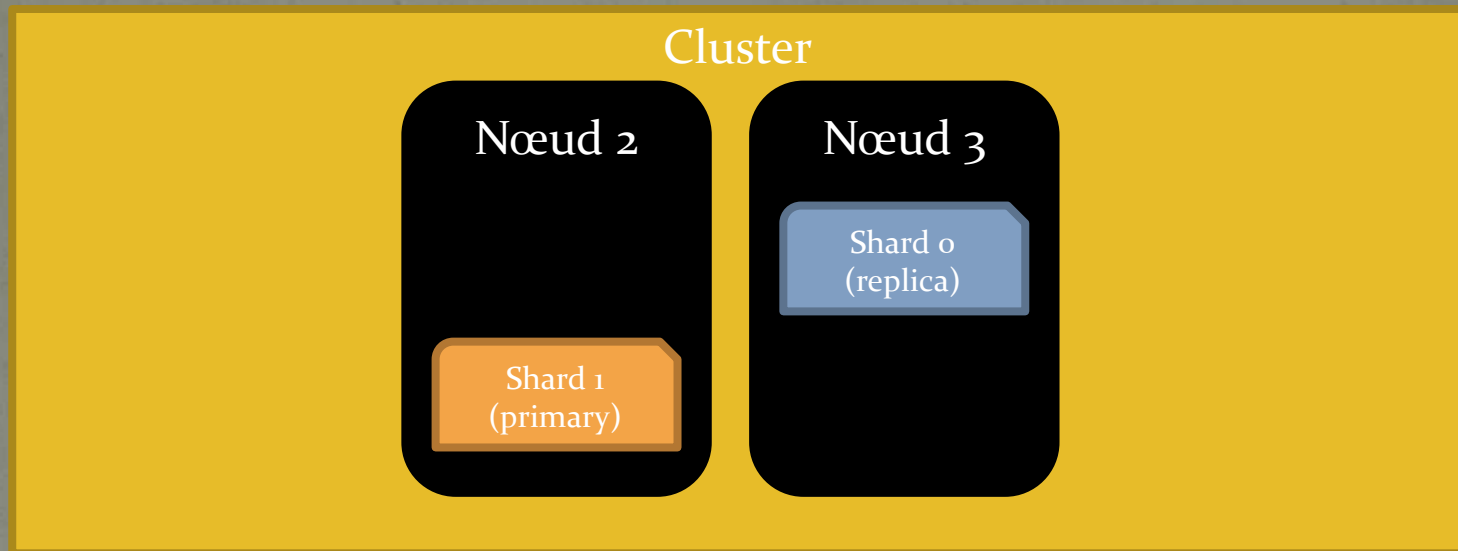
Ajoutons un nœud



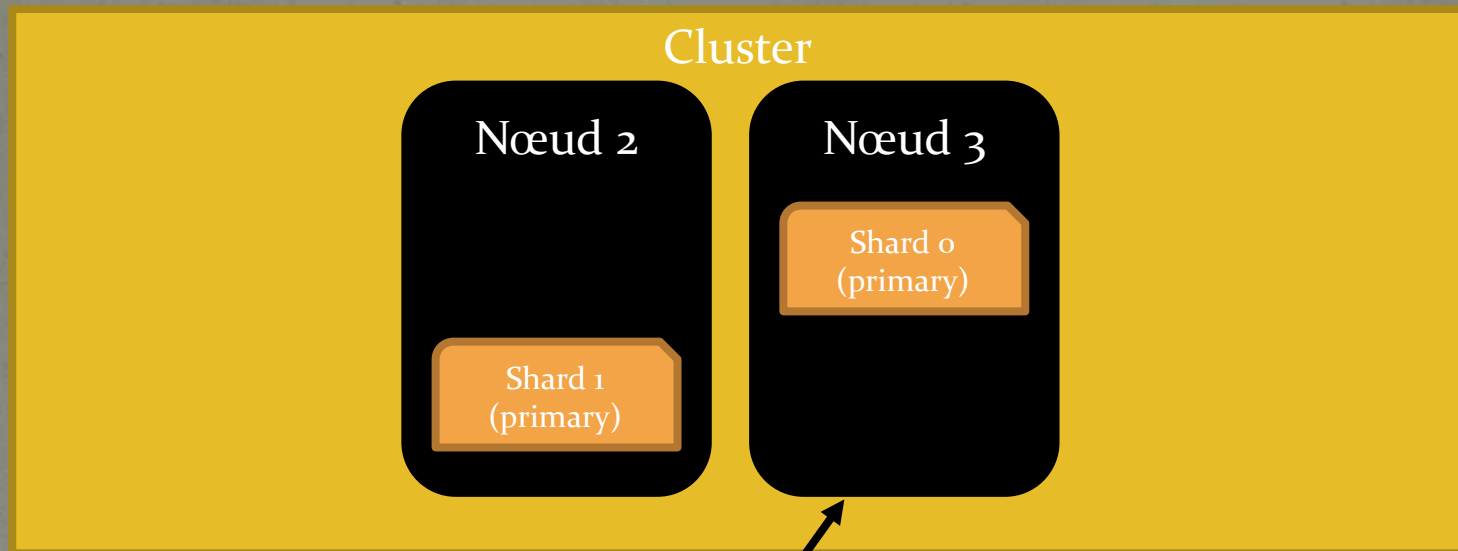
Ajoutons un nœud



Aïe ! Le nœud 1 disparaît !



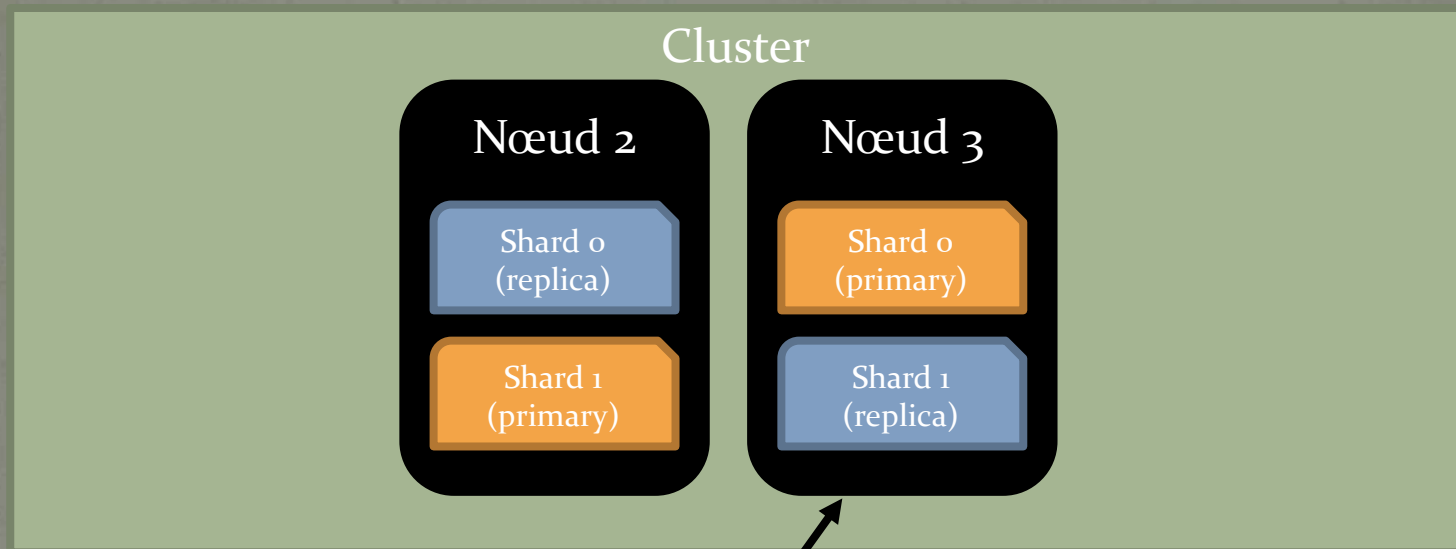
Aïe ! Le nœud 1 disparaît !



Shard 0 / nœud 3 devient
primary



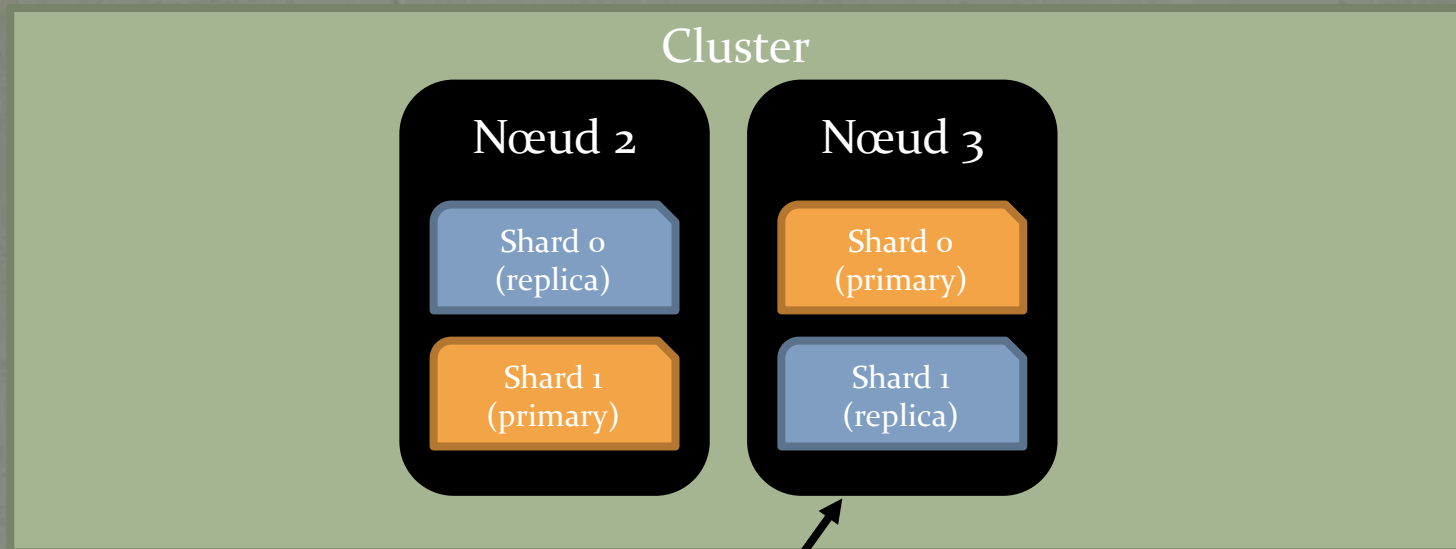
Aïe ! Le nœud 1 disparaît !



Cluster à nouveau **GREEN**



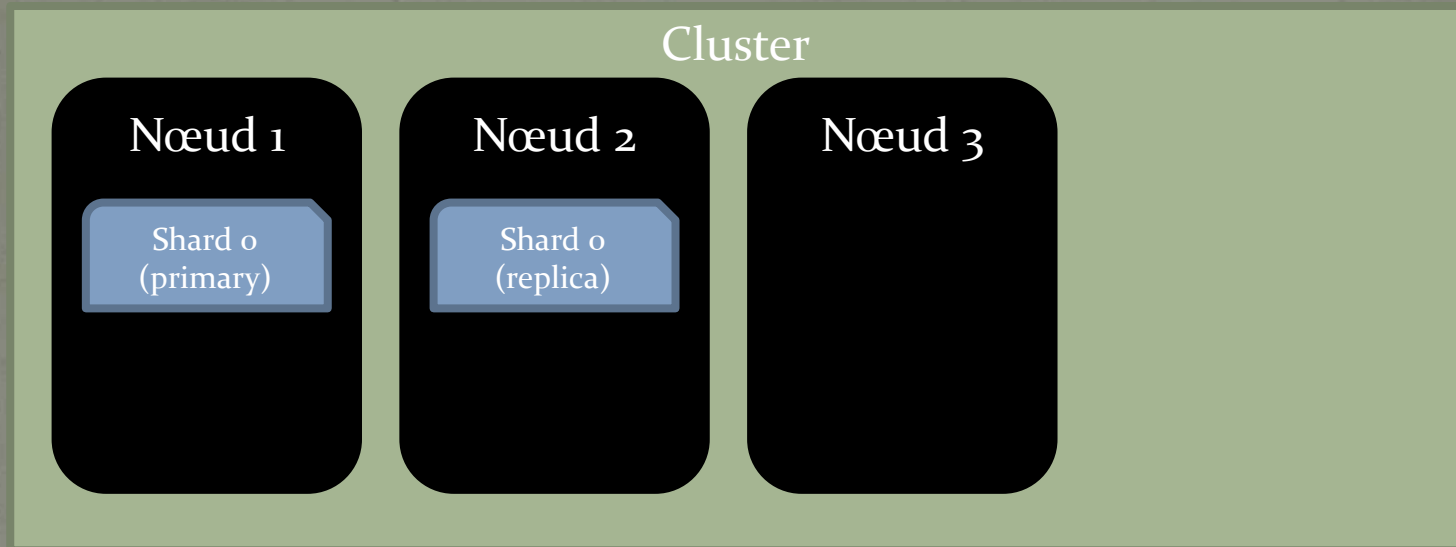
Aïe ! Le nœud 1 disparaît !



Cluster à nouveau **GREEN**



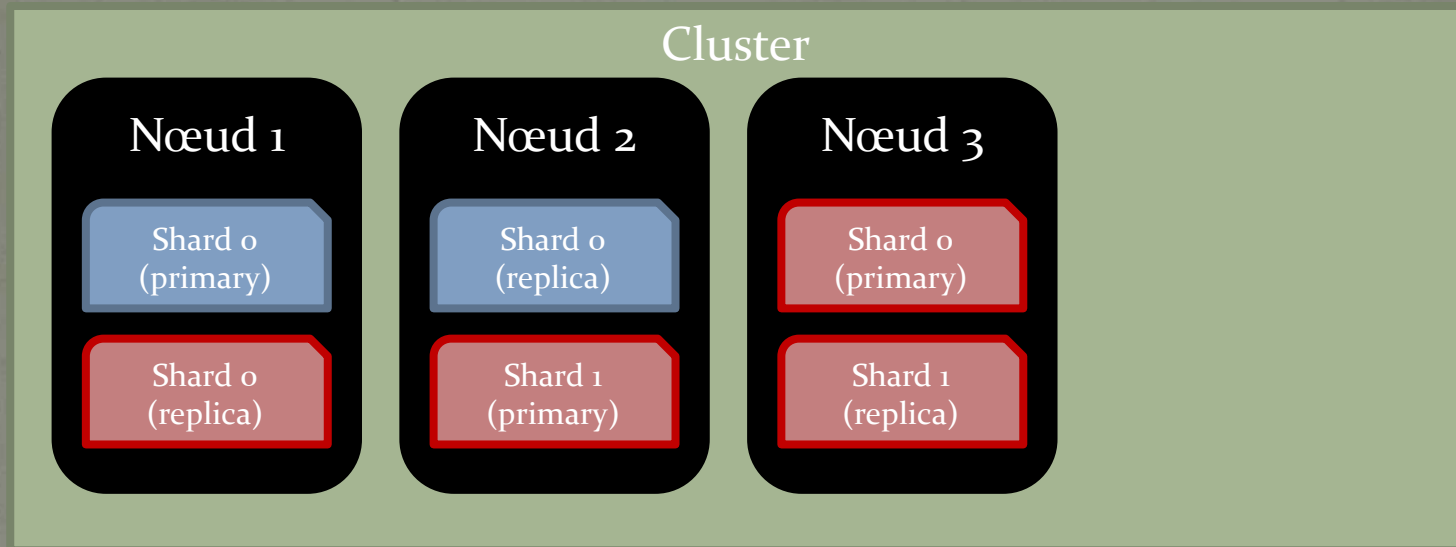
Gestion multi-index



```
$ curl -XPUT localhost:9200/test1 -d '{  
  "index" : {  
    "number_of_shards" : 1,  
    "number_of_replicas" : 1  
  }  
'
```



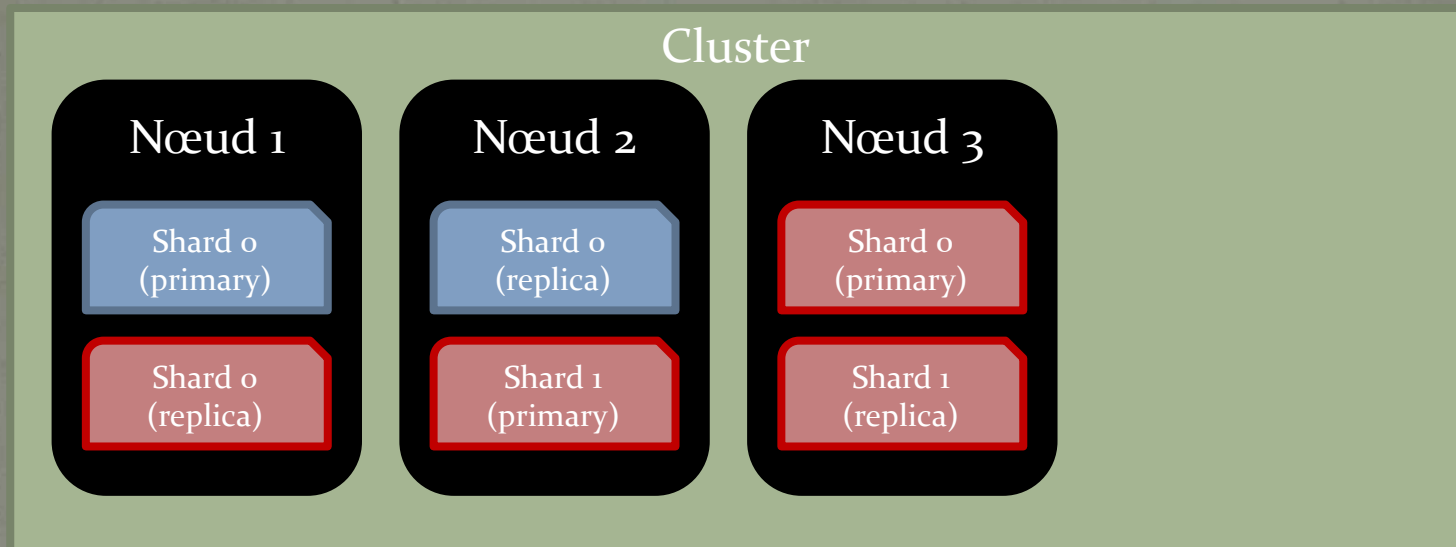
Gestion multi-index



```
$ curl -XPUT localhost:9200/test2 -d '{
  "index" : {
    "number_of_shards" : 2,
    "number_of_replicas" : 1
  }
}'
```



Recherche multi index



```
$ curl localhost:9200/test1/_search
$ curl localhost:9200/test1,test2/_search
$ curl localhost:9200/_search

$ TODO CREATE ALIAS for TESTS on TEST1,TEST2
$ curl localhost:9200/tests/_search
```


Les recherches

```
$ curl localhost:9200/test1/_search  
$ curl localhost:9200/test1,test2/_search  
$ curl localhost:9200/_search  
  
$ TODO CREATE ALIAS for TESTS on TEST1,TEST2  
$ curl localhost:9200/tests/_search
```

TODO



Les facettes

```
$ curl localhost:9200/test1/_search  
$ curl localhost:9200/test1,test2/_search  
$ curl localhost:9200/_search  
  
$ TODO CREATE ALIAS for TESTS on TEST1,TEST2  
$ curl localhost:9200/tests/_search
```

TODO



Le mapping

```
$ curl localhost:9200/test1/_search  
$ curl localhost:9200/test1,test2/_search  
$ curl localhost:9200/_search  
  
$ TODO CREATE ALIAS for TESTS on TEST1,TEST2  
$ curl localhost:9200/tests/_search
```

TODO



Les analyseurs

```
$ curl localhost:9200/test1/_search  
$ curl localhost:9200/test1,test2/_search  
$ curl localhost:9200/_search  
  
$ TODO CREATE ALIAS for TESTS on TEST1,TEST2  
$ curl localhost:9200/tests/_search
```

TODO



Percolation

```
$ curl localhost:9200/test1/_search  
$ curl localhost:9200/test1,test2/_search  
$ curl localhost:9200/_search  
  
$ TODO CREATE ALIAS for TESTS on TEST1,TEST2  
$ curl localhost:9200/tests/_search
```

TODO



Rivers

```
$ curl localhost:9200/test1/_search  
$ curl localhost:9200/test1,test2/_search  
$ curl localhost:9200/_search  
  
$ TODO CREATE ALIAS for TESTS on TEST1,TEST2  
$ curl localhost:9200/tests/_search
```

TODO

