elasticsearch介绍与使用

-- 周志超

2017.03.20

JD.京东.COM

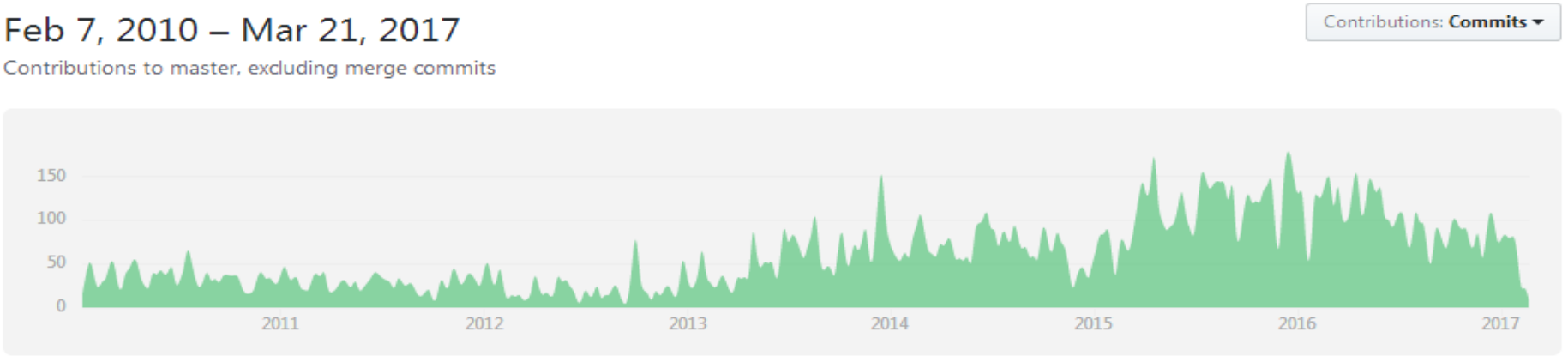
目录

JD.COM 京东

1. elasticsearch介绍

JD.COM 京东

Elasticsearch 是一个基于Lucene的分布式、可扩展、近实时的搜索与数据分析引擎。 它能从项目一开始就赋予你的数据以搜索、分析和探索的能力. 实时搜索，稳定，可靠，快速，安装使用方便。Elasticsearch 可以横向扩展至数百（甚至数千）的服务器节点，同时可以处理PB级数据。目前版本：v5.2.2



官方网站： https://www.elastic.co/

文档：https://www.elastic.co/guide/en/elasticsearch/reference/current/getting-started.html

中文社区：http://elasticsearch.cn/

中文翻译：http://106.186.120.253/preview/foreword_id.html

2.基本概念

JD.COM 京东

ES vs **DB**

MySQL	elasticsearch
database(数据库)	index(索引库)
table(表)	type(类型)
row(行)	document(文档)
column(列)	field(字段)

Keyword:

cluster 集群

node 节点

index 索引

type 类型

document 文档

shards & replicas 分片&副本

field 字段,域

setting 设置

mapping 映射

template 模板

2.基本概念

JD.COM 京东

cluster

集群是一个或多个节点（服务器）的集合，它们联合起来保存所有的数据（索引以分片为单位分散到多个节点上保存）并且可以在所有的节点上进行索引和搜索操作。默认的名字是“elasticsearch”。最好指定一个自己的名字，比如“jiesi-123”。用户通过集群名，ip，port访问集群。

node

节点是一个服务器，也是集群的一部分，它存储你的数据，并参与集群的索引和搜索。节点通过配置集群的名称，就可以加入到集群中。默认，节点都加入一个叫elasticsearch的集群。

index

索引是很多文档的集合，这些文档都具备一些相似的特征。例如，你可以创建一个存放商品信息的product索引，存放用户信息的user索引。(名词：an index of es。动词：index a document to es)

Type

你可以在索引中定义一个或多个类型。类型是索引的一个逻辑分类或划分，它的概念完全取决于你自己如何去理解。通常，类型是为一些具备公共字段的文档定义的。例如，有一个person索引，在这个索引下定义teacher，student两个类型。如果两个类型的字段集互不相同，这意味着索引中将有一半的数据是空的，最终将导致性能问题。

document

文档是可以被索引的基本单位。文档使用JSON格式。比如：{“name”：“李雷”，“age”:16}。

文档本质上是存放在索引中，但实际上是被索引到索引中的一个类型中。

2.基本概念

JD.COM 京东

shards & replicas

elasticsearch 中索引数据被划分到多个分片上，分片保存到不同的节点（服务器）上。创建索引时可以指定分片和副本数量。默认5个主分片和1一个复制分片。可以动态地改变复制分片的数量，但不能改变主分片的数量。每个elasticsaerch分片都是一个Lucene 索引。在单个索引中你最多存储2,147,483,519 个文档。

分片的主要是为了以下两个目的：

- 水平划分/扩展你的内容
- 并行地分发操作到多个节点的分片上，提升性能或吞吐量。

复制分片主要是为了以下两个目的：

- 提供高可用性，以防节点/分片不可以用。不要把复制分片和主分片放在同一个节点上。
- 扩大搜索量和吞吐量，因为不同的搜索请求可以在所有的复制分片中并发执行。

field

field就是要存在ES中的文档的字段，一个java对象的成员变量，相当于数据库中的列。比如文档：{“name”：“李雷”，“age”:16}。Name，age就是field。

2.基本概念

JD.COM 京东

settings

settings是对于index的配置。比如几个分片，几个副本，刷新频率等。创建索引时指定,某些值也可以动态更新。

```
PUT /my_temp_index
{
    "settings": {
        "number_of_shards" :   5,
        "number_of_replicas" : 1,
        "refresh_interval":  “1s"
    }
}
```

用 update-index-settings API 动态修改setting：

```
PUT /my_temp_index/_settings
{
    "number_of_replicas": 1
}
```

2.基本概念

JD.COM 京东

mapping

为了能够将时间域视为时间，数字域视为数字，字符串域视为全文或精确值字符串， Elasticsearch 需要知道每个域中数据的类型。这个信息包含在mapping中。mapping是索引中存放的数据结构类型的定义，对索引库中索引的字段名称及其数据类型进行定义，类似于关系数据库中表建立时要定义字段名及其数据类型那样。

es的mapping比数据库灵活很多，它可以动态添加字段。一般不需要要指定mapping都可以，因为es会自动根据数据格式定义它的类型，如果你需要对某些字段添加特殊属性（如：定义使用其它分词器、是否分词、是否存储等），就必须手动添加mapping。定义了文档的每个field的数据类型，是否索引，是否存储，格式，分词器等信息。当你索引一个包含新域的文档--之前未曾出现-- Elasticsearch 会使用 动态映射。

举个栗子：

```
{
  "userName": "赵日天",
  "cardId": "75987134578",
  "operate_time": "2017-3-23 10:55:00"
}
```

```
{
  "mappings": {
    "my_type_name": {
      "properties": {
        "userName": {
          "type": "string",
          "index": "not_analyzed"
        },
        "cardId": {
          "type": "long"
        },
        "operate_time": {
          "type": "date",
          "format": "yyyy-MM-dd HH:mm:ss"
        }
      }
    }
  }
}
```

2.基本概念

JD.COM 京东

template

template是mapping的一个模板。当创建索引时，如果索引名字满足模板的规则，模板定义的mapping将直接应用到该索引。如果每天都需要创建索引的话，定义好模板之后，创建索引将是一件异常简单的事。

如果我创建索引 es_log20170323,template中定义的settings，mapping将自动应用到改索引中。

```
{
  "template": "es_log*",
  "settings": {
    "index": {
      "number_of_shards": "8",
      "number_of_replicas": "0"
    }
  },
  "mappings": {
    "app_log": {
      "properties": {
        "AppVersion": {
          "index": "not_analyzed",
          "type": "string"
        },
        "Count": {
          "type": "integer"
        },
        "RequestTime": {
          "type": "date"
        }
      }
    }
  }
}
```

2.基本概念

JD.COM 京东

Elasticsearch 支持 如下简单域类型：

- 字符串: string
- 整数 : byte, short, integer, long
- 浮点数: float, double
- 布尔型: boolean
- 日期: date

```
{
  "properties": {
    "birthday": {
      "type": "date",
      "format": "yyyy-MM-dd",
      "index": "no"
    },
    "comment": {
      "type": "string",
      "analyzer": "english",
      "store": "no"
    }
  }
}
```

index属性用于控制字段是否被索引，以及如何索引？可以配置为下列任一值：

- analyzed: 首先分析字符串，然后索引它。换句话说，以全文索引这个域。默认值。
- not_analyzed: 索引这个域，所以可以搜索到它，但索引指定的精确值。不对它进行分析。
- no: 不索引这个域,不会被搜索到。

不需要分析的字段别分析。Id之类的不分析，

store

有yes、 no 两种取值，来指定字段值是否保存到索引区。默认设置为no，该字段将不会保存至索引区。

analyzer

为建索引的字段，指定分词器。比如：english,standard,ik

2.基本概念

JD.COM 京东

访问集群

Ip : 192.168.200.190

Tcp port : 9303

http port : 9203

Curl发送一个 Elasticsearch 请求：

curl -X<method> ‘http://ip:port/<PATH>?<QUERY_STRING>' -d '<BODY>'

method: GET，POST，PUT，HEAD ，DELETE

PATH:API 的终端路径,Path 可能包含多个组件，

例如：

curl –XGET 192.168.200.190:9203/_cluster/stats

curl –XGET 192.168.200.190:9203/_nodes/stats

curl -XGET 192.168.200.190:9203/index_name/my_type_name/_search?q=age:18

curl -XPOST 192.168.200.190:9203/index_name/my_type_name/1 -d '{"name":"李雷","age":16} ‘

curl -XPUT 192.168.200.190:9203/index_name

curl -XDELETE 192.168.200.190:9203/index_name

curl -XPOST 192.168.200.190:9203/index_name/_mapping/my_type_name -d '{"properties":{"name":{"type":"string"},"age":{"type":"integer"}}} ‘

3.API使用

JD.COM 京东

访问集群

[root@200_190 download]# **curl 192.168.200.190:9203**


```
{
  "name" : "gw_192.168.200.190:9203",
  "cluster_name" : "jiesi-1",
  "version" : {
    "number" : "2.1.2",
    "build_hash" : "63c285e4741baf609e963cb9d463ad1a47e1a179",
    "build_timestamp" : "2016-01-27T12:57:52Z",
    "build_snapshot" : false,
    "lucene_version" : "5.3.1"
  },
  "tagline" : "You Know, for Search"
}
```


创建索引

curl -XPUT 192.168.200.190:9203/index_name

创建mapping

curl -XPOST 192.168.200.190:9203/index_name/_mapping/my_type_name -d

'{"properties":{"name":{"type":"string"},"age":{"type":"integer"}}}'

3.API使用

JD.COM 京东

创建索引，mapping

```
curl -XPOST 192.168.200.190:9203/index_name -d  '{
  "mappings": {
    "my_type_name": {
      "properties": {
        "name": {
          "type": "string"
        },
        "age": {
          "type": "integer"
        }
      }}}
} ‘
```

修改mapping

curl -XPOST 192.168.200.190:9203/index_name/_mapping/my_type_name -d '{"properties":{"description":{"type":"string"}}} ‘

修改mapping只能新增field，对于已有field不能修改！！！

插入文档：

curl -XPUT 192.168.200.190:9203/index_name/my_type_name/1 -d '{"name":"李雷","age":16}'

curl -XPUT 192.168.200.190:9203/index_name/my_type_name/2 -d '{"name":"韩梅梅","age":18} ‘

POST自动生成id

curl -XPOST 192.168.200.190:9203/index_name/my_type_name -d '{"name":"韩梅梅","age":18} ‘

3.API使用

JD.COM 京东

部分更新：

```
curl -XPOST 192.168.200.190:9203/index_name/my_type_name/1/_update -d
' {
  "doc": {
    "name": "李小雷"
  }
} '
```

elasticsearch底层实际上并没有更新文档。当我们执行更新操作时，elasticsearch删除旧的文档然后索引新的文档。

覆盖：

```
curl -XPUT 192.168.200.190:9203/index_name/my_type_name/1 -d
  ‘{
  "name": "李雷2",
  "age": 26
}’
```

3.API使用

JD.COM 京东

批量操作

curl -XPOST 192.168.200.190:9203/index_name/my_type_name/_bulk -d '

{"index":{"_id":"3"}}

{"name": "李华" }

{"index":{"_id":"4"}}

{"name": "苏珊" }

{"delete":{"_id":"1"}}

{"update":{"_id":"2"}}

{"doc":{"name":"韩小梅"}}

'

或：curl -XPOST 192.168.200.190:9203/index_name/my_type_name/_bulk --data-binary @test.json

批量操作会按顺序的执行所有操作。如果某个操作执行失败，它会继续执行剩下的操作。api返回结果时，会提供每一个操作的状态（和操作的顺序一致），你可以通过这个状态检查操作是否执行成功。

查看一下结果

curl -XGET 192.168.200.190:9203/index_name/my_type_name/_search

返回：

{"_shards":{"total":5,"failed":0,"successful":5},"hits":{"hits":[{"_index":"index_name","_type":"my_type_name","_source":{"name":"苏珊"},"_id":"4","_score":1},{"_index":"index_name","_type":"my_type_name","_source":{"name":"韩小梅","age":18},"_id":"2","_score":1},{"_index":"index_name","_type":"my_type_name","_source":{"name":"李华"},"_id":"3","_score":1}],"total":3,"max_score":1},"took":3,"timed_out":false}

3.API使用

JD.COM 京东

curl -XGET 192.168.200.190:9203/index_name/my_type_name/_search


```
{"hits": {
    "hits": [
     {
       "_index": "index_name",
       "_type": "my_type_name",
       "_source": {
         "name": "李雷",
         "age": 16
       },
       "_id":   "1","_score": 1
     },
     {
       "_index": "index_name",
       "_type": "my_type_name",
       "_source": {
         "name": "韩梅梅",
         "age": 18
       },
       "_id":   "2","_score": 1
     }
    ],
    "total": 2,"max_score": 1
  },
  "took": 4,"timed_out": false}
```


通过URI传递查询参数：

curl -XGET 192.168.200.190:9203/index_name/my_type_name/_search ? q=*&size=10&sort=age:asc

curl -XGET 192.168.200.190:9203/index_name/my_type_name/_search?q=age:18

3.API使用

JD.COM 京东

通过request body发送请求参数：

curl -XGET 192.168.200.190:9203/index_name/my_type_name/_search –d

```
‘{
 "size": 10,
 "query": {
  "range": {
   "age": {
    "gte": 18,
    "lte": 60
   }
  }
 },
 "sort": [
  {
   "age": "asc"
  }
 ]
}
’
```

Term curl -XPOST 192.168.200.190:9203/index_name/_search?pretty -d '{"query":{"term":{"age":16}}}'

Match curl -XPOST 192.168.200.190:9203/index_name/_search?pretty -d '{"query":{"match":{"name":"李雷"}}}'

Range curl -XPOST 192.168.200.190:9203/index_name/_search?pretty -d '{"query":{"range":{"age":{"gte":18,"lte":60}}}} ‘

Prefix curl -XPOST 192.168.200.190:9203/index_name/_search?pretty -d '{"query":{"prefix":{"name":"李"}}}'

Bool curl -XPOST 192.168.200.190:9203/index_name/_search?pretty -d ‘{"query":{"bool":{"must":{"match":{"name":"李雷和韩梅梅
"}},"filter":{"range":{"age":{"from":10,"to":20}}},"must_not":{"term":{"name":"张
"}},"should":[{"term":{"age":16}},{"term":{"age":18}}],"minimum_should_match":1,"boost":1}}’

3.API使用

JD.COM 京东

获取文档：

curl -XGET 192.168.200.190:9203/index_name/my_type_name/1

```
{
  "found": true,
  "_index": "index_name",
  "_type": "my_type_name",
  "_source": {
     “name” : “李雷",
    "age": 18
  },
  "_id": “1",
  "_version": 1
}
```

删除文档：

curl -XDELETE 192.168.200.190:9203/index_name/my_type_name/1

删除索引index_name：

curl -XDELETE 192.168.200.190:9203/index_name

3.API使用

JD.COM 京东

总结

创建索引：

curl -XPOST 192.168.200.190:9203/index_name

增：curl -XPUT 192.168.200.190:9203/index_name/my_type_name/1 -d '{"name":"李雷","age":16}'

删：**curl -XDELETE 192.168.200.190:9203/index_name/my_type_name/1**

改：curl -XPOST 192.168.200.190:9203/index_name/my_type_name/1/_update -d ' {"doc":{"name":"李小雷"}} '

查：**curl -XGET 192.168.200.190:9203/index_name/my_type_name/1**

删除索引：

curl -XDELETE 192.168.200.190:9203/index_name

3.API使用

JD.COM 京东

集群信息查看

curl -XPOST 192.168.200.190:9203 /_cluster/stats

The Cluster Stats API allows to retrieve statistics from a cluster wide perspective. The API returns basic index metrics (shard numbers, store size, memory usage) and information about the current nodes that form the cluster (number, roles, os, jvm versions, memory usage, cpu and installed plugins).

curl -XPOST 192.168.200.190:9203 /_cluster/state

The cluster state API allows to get a comprehensive state information of the whole cluster.

metadata

集群索引的settings，mapping，template等元数据

nodes

节点信息

routing_nodes

节点维度 节点上有哪些索引的分片

routing_table

索引维度 索引的分片分别分配到哪些节点


```
{
  "cluster_name": "jiesi-1",
  "metadata": {
    "indices": {},
    "cluster_uuid": "wTbSO96JSnOkGxOMIgsCyg",
    "templates": {}
  },
  "nodes": {},
  "blocks": {},
  "routing_nodes": {},
  "routing_table": {},
  "state_uuid": "pUAzDXYTRX6Y5qgc-DLlQg",
  "master_node": "JTYdQKdNTBurFbWGW9WvBQ",
  "version": 347
}
```

3.API使用

JD.COM 京东

查看集群健康状态

[root@200_190 download]# **curl -XGET 192.168.200.190:9203/_cat/health?v**

```
epoch      timestamp cluster status node.total node.data shards pri relo init unassign pending_tasks max_task_wait_time active_shards_percent
1490155141 11:59:01  jiesi-1 red          10         5   1119 556    0    0       8             0                  -                  99.3%
```

green:一切良好（集群所有的功能都正常）。

yellow:所有的数据都是可用的，但是一些复制分片可能没有正确分发（集群的所有功能还正常）。

red:主分片丢失，有些数据不能使用。它仍然可以运行一部分的功能。（它依然可以从一些可用的分片处理搜索请求）应该尽快去修复它。

[root@200_190 download]# **curl -XGET 192.168.200.190:9203/_cat/nodes?v**

```
host            ip              heap.percent ram.percent load node.role master name
192.168.200.196 192.168.200.196            5          99 0.12 d         *      dm_192.168.200.196:9203
192.168.200.190 192.168.200.190            5          99 0.22 -         -      gw_192.168.200.190:9203
192.168.200.191 192.168.200.191            1          92 0.15 -         -      gw_192.168.200.191:9203
192.168.200.195 192.168.200.195            5          99 0.00 d         m      dm_192.168.200.195:9203
192.168.200.198 192.168.200.198           12          99 0.00 d         -      d_192.168.200.198:9203
192.168.200.197 192.168.200.197            9          99 0.87 d         -      d_192.168.200.197:9203
192.168.200.189 192.168.200.189            5          82 0.11 -         -      gw_192.168.200.189:9203
192.168.200.194 192.168.200.194            6          99 0.15 d         m      dm_192.168.200.194:9203
192.168.200.193 192.168.200.193            5          93 0.16 -         -      gw_192.168.200.193:9203
192.168.200.192 192.168.200.192            4          99 1.21 -         -      gw_192.168.200.192:9203
```

3.API使用

JD.COM 京东

查看分片分配情况

```
[root@200_190 download]# curl 192.168.200.190:9203/_cat/allocation?v
shards disk.indices disk.used disk.avail disk.total disk.percent host           ip             node
  1039      718.5mb    25.6gb     84.5gb    110.2gb           23 192.168.200.194 192.168.200.194 dm_192.168.200.194:9203
  1023      683.2mb    41.2gb       69gb    110.2gb           37 192.168.200.198 192.168.200.198 d_192.168.200.198:9203
  1080      512.6mb    40.1gb     70.1gb    110.2gb           36 192.168.200.195 192.168.200.195 dm_192.168.200.195:9203
  1437        1.3gb    70.3gb     39.8gb    110.2gb           63 192.168.200.196 192.168.200.196 dm_192.168.200.196:9203
  1064     1013.5mb    63.6gb     46.5gb    110.2gb           57 192.168.200.197 192.168.200.197 d_192.168.200.197:9203
    35                                                                             UNASSIGNED
```

查看索引

```
[root@200_190 download]# curl '192.168.200.190:9203/_cat/indices?bytes=b' | sort -rnk8|head -10
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 34749  100 34749    0     0  72198      0 --:--:-- --:--:-- --:--:-- 72393
green open .node_monitor_jiesi-1-2017.03.20                1 1  758975     0 496971403 247940757
green open .node_monitor_jiesi-1-2017.03.19                1 1  941812     0 450934542 225467271
green open pop-bean                                        1 0 2255116 58612 200456887 200456887
green open metrics_alias1                                  5 0 1238799     0 189294666 189294666
green open .node_monitor_es-ucc-2017.03.18                 1 1  306300     0 187138720  93569360
green open .node_monitor_es-ucc-2017.03.19                 1 1  306115     0 184821644  92410822
green open form_index                                      5 1  603783   736 173567711  86777763
green open .node_monitor_es-ucc-2017.03.20                 1 1  230487     0 151279058  75640750
green open kdc_es                                         10 1   80683 14773 129936606  64968303
green open app_dashboard_report_sku_dc_main_detail_dc     16 1  389856     0 128751618  64375809
```

3.API使用

JD.COM 京东

cat支持的API

```
[root@200_190 download]# curl 192.168.200.190:9203/_cat
=^.^=
/_cat/allocation
/_cat/shards
/_cat/shards/{index}
/_cat/master
/_cat/nodes
/_cat/indices
/_cat/indices/{index}
/_cat/segments
/_cat/segments/{index}
/_cat/count
/_cat/count/{index}
/_cat/recovery
/_cat/recovery/{index}
/_cat/health
/_cat/pending_tasks
/_cat/aliases
/_cat/aliases/{alias}
/_cat/thread_pool
/_cat/plugins
/_cat/fielddata
/_cat/fielddata/{fields}
/_cat/nodeattrs
/_cat/repositories
/_cat/snapshots/{repository}
```

3.API使用

JD.COM 京东

_refresh

写入和打开一个新段的轻量的过程。 默认情况下每个分片会每秒自动刷新一次。这就是为什么我们说 Elasticsearch 是 近 实时搜索: 文档的变化并不是立即对搜索可见，但会在一秒之内变为可见。你的应用需要意识到 Elasticsearch 的近实时的性质，并接受它的不足.

POST my_index /_refresh //手动执行一次refresh，使搜索马上可见

PUT my_index {“settings”:{“refresh_interval”:“30s” }} //值-1表示不刷新。 注意单位

_flush

执行一个提交并且截断 translog 。 分片每30分钟，或者在 translog 太大的时候自动刷新（flush）。你很少需要自己执行一个手工的 `flush`；通常情况下，自动刷新就足够了。在重启节点或关闭索引之前执行 flush 有益于你的索引。当 Elasticsearch 尝试恢复或重新打开一个索引， 它需要重放 translog 中所有的操作，所以如果日志越短，恢复越快。

POST my_index/_flush //手动执行刷硬盘操作，持久化变更。

一个文档的旅行：

内存--（_refresh）-->**文件系统缓存**--（_flush）-->**硬盘**

不能被搜索　　可搜索　　持久化

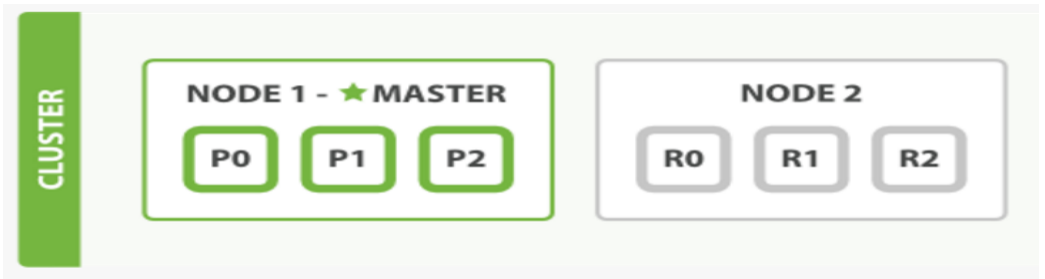
4.节点分片

JD.COM 京东

3 primary shards

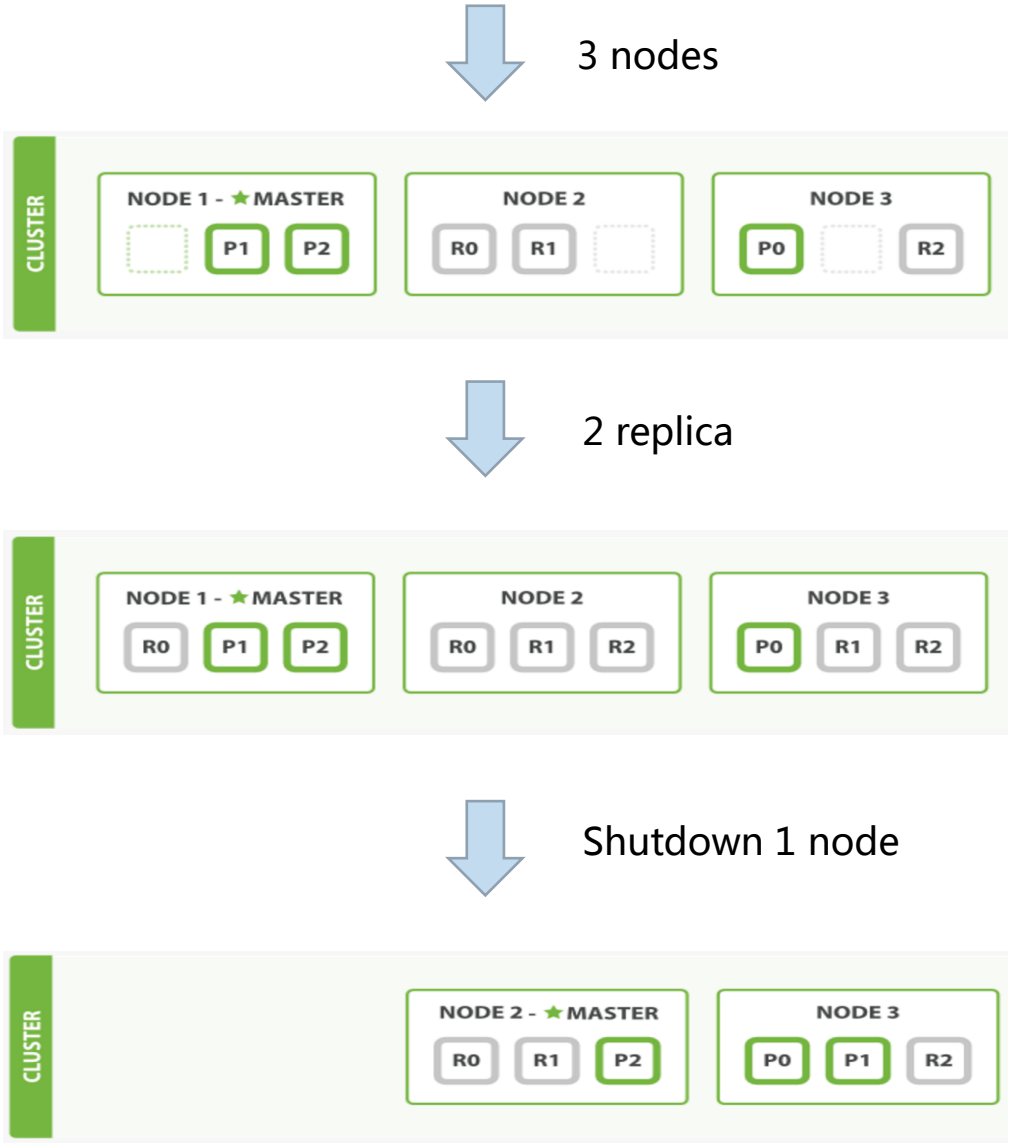
1 replica

2 nodes



4.节点分片

JD.COM 京东



reference：

《Elasticsearch Reference（2.1）》https://www.elastic.co/guide/en/elasticsearch/reference/2.1/getting-started.html
《The Definitive Guide》https://www.elastic.co/guide/en/elasticsearch/guide/current/index.html
《Elasticsearch基础教程》http://blog.csdn.net/cnweike/article/details/33736429
《Elasticsearch 权威指南》http://es.xiaoleilu.com/index.html
《Elasticsearch权威指南翻译目录》http://blog.csdn.net/dm_vincent/article/details/46994535/
中文权威指南：http://106.186.120.253/preview/geoloc.html

谢 谢 ！