

# Elasticsearch集群运维实战

## 第0章 课程面向人群

### 1.适合哪些人

1. 运维人员或者准备从事运维工作的人员
2. 需要维护Elasticsearch数据库的IT人员
3. 想学习和使用ELK日志分析的IT人员

### 2.不适合哪些人

1. 想了解Elasticsearch的详细底层原理的DBA或者开发人员。
2. 想了解Elasticsearch在具体的开发项目中代码如何实现或者编写的开发人员。

### 3.课程目标

1. 不涉及到操作系统或者java语言的底层原理
2. 不涉及到具体的业务代码
3. 学完之后可以独立部署和维护Elasticsearch集群的稳定运行
4. 为后面的ELK课程做铺垫打基础

## 第1章 Elasticsearch介绍

### 1.什么是Lucene

1. Lucene是一个高性能的java搜索引擎,操作非常繁琐,需要具备java开发经验。
2. Elasticsearch是基于Lucene之上包装一层外壳,屏蔽了Lucene的复杂操作,即使不会java语言也可以快速上手。

### 2.什么是全文检索和倒排索引

#### 2.1 什么是索引

1. 索引就好比书的目录,如果我们想快速查看某个章节,只需要找到目录里相应章节对应的页数即可。
2. 通过目录找到章节,通过章节找到页码这个过程就是索引的过程。
3. 索引的目的就是加快数据搜索的效率。

#### 2.2 什么是全文检索

1. 先建立索引,再对索引进行搜索的过程就叫全文检索(Full-text Search)。

## 2.3 什么是倒排索引

- 1 索引是根据章节找到页数，但是如果我并不知道我要找的内容属于哪个章节，比如我只知道一个关键词，但是不知道这个关键词属于哪个章节。大家可以想一下，我们平时利用搜索引擎搜索的时候是不是也是这种场景呢？
- 2 比如我们想知道一个电影的名字，但是记不起来具体的名字，只知道部分关键词或者剧情的内容，那这种情景背后如何用技术解决呢？
- 3 这时候就不得不提到倒排索引了。
- 4
- 5 那么什么是倒排索引呢？还是拿书的目录举例子：
- 6 正常索引：
- 7 第1章 Elasticsearch介绍 第10页
- 8 第2章 Elasticsearch安装配置 第15页
- 9 第3章 Elasticsearch自定义配置 第20页
- 10
- 11 倒排索引：
- 12 关键词 章节
- 13 Elasticsearch 第1章 第2章 第3章
- 14 安装 第2章
- 15 配置 第2章 第3章
- 16 自定义 第3章

再举一个例子：

- 1 假设数据里有以下新闻标题：
- 2 1.老男孩教育 1
- 3 2.老男孩教育linux学院 1 1 == 2
- 4 3.老男孩教育python学院 1 1 == 2
- 5 4.老男孩教育DBA学院 1 1 == 3
- 6 5.老男孩教育oldzhang 1
- 7 6.老男孩教育安全 1
- 8
- 9 Elasticsearch 内部 分词,评分,倒排索引：
- 10 老男孩 1 2 3 4 5 6
- 11 教育 1 2 3 4 5 6
- 12 学院 2 3
- 13 linux 2
- 14 python 3
- 15 DBA 4
- 16 安全 6
- 17
- 18 用户输入：
- 19 老男孩 学院
- 20 老男孩学院DBA
- 21
- 22 ES：
- 23 老男孩
- 24 学院
- 25 linux
- 26 DBA

## 3.Elasticsearch应用场景

- 1 1.搜索： 电商,百科,app搜索,搜索结果高亮显示
- 2 2.日志分析和数据挖掘,数据展示

## 4.Elasticsearch特点

1. 高性能,天然分布式集群
2. 对运维友好,不需要会java语言,开箱即用,配置文件精简
3. 功能丰富,社区活跃,版本更新特别的快 2.6 -- 8.0

## 5.Elasticsearch在电商搜索的实现

```
1 cat
2 skuid  name
3 1      狗粮100kg
4 2      猫粮50kg
5 3      猫罐头200g
6
7 select * from cat where name like '%'
8
9 Elasticsearch:
10 聚合运算之后得到SKUID:
11 1
12 2
13
14 拿到ID之后,mysql就只需要简单地where查询即可
15 mysql:
16 select xx from xxx where skuid 1
```

## 第2章 Elasticsearch安装

### 1.关闭防火墙和Selinux

```
1 关闭swap分区
2 内存 2G
3
4 iptables -nL
5 iptables -F
6 iptables -X
7 iptables -Z
8 iptables -nL
9
10 #关闭selinux
11 临时生效:
12 setenforce 0
13 getenforce
14
15 永久生效:
16 setenforce 0
17 vim /etc/selinux/config
18 SELINUX=disabled
```

### 2.下载软件

```
1 mkdir /data/soft -p
2 cd /data/soft/
3 wget https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-7.9.1-x86_64.rpm
```

## 3.安装jdk

1 对于Elasticsearch7.0之后的版本不需要再独立的安装JDK了，软件包里已经自带了最新的JDK，所以直接启动即可。

## 3.安装ES

```
1 rpm -ivh elasticsearch-7.9.1-x86_64.rpm
```

## 4.启动并检查

```
1 systemctl daemon-reload
2 systemctl enable elasticsearch.service
3 systemctl start elasticsearch.service
4 netstat -lntup|grep 9200
5 curl 127.0.0.1:9200
```

# 第3章 Elasticsearch自定义配置

## 1.查看ES有哪些配置

```
1 [root@node-51 ~]# rpm -qc elasticsearch
2 /etc/elasticsearch/elasticsearch.yml          #主配置文件
3 /etc/elasticsearch/jvm.options                #JVM配置文件
4 /etc/init.d/elasticsearch                      #init
   启动脚本
5 /etc/sysconfig/elasticsearch                  #环境变量文
   件
6 /usr/lib/sysctl.d/elasticsearch.conf          #内核参数文件
7 /usr/lib/systemd/system/elasticsearch.service #systemd启动文件
```

## 2.自定义配置文件

```
1 cp /etc/elasticsearch/elasticsearch.yml /opt/
2 cat > /etc/elasticsearch/elasticsearch.yml << 'EOF'
3 node.name: node-1
4 path.data: /var/lib/elasticsearch
5 path.logs: /var/log/elasticsearch
6 bootstrap.memory_lock: true
7 network.host: 127.0.0.1,10.0.0.51
8 http.port: 9200
9 discovery.seed_hosts: ["10.0.0.51"]
10 cluster.initial_master_nodes: ["10.0.0.51"]
11 EOF
```

配置文件解释:

```
1 node.name: node-1 #节点名称
2 path.data: /var/lib/elasticsearch #数据目录
3 path.logs: /var/log/elasticsearch #日志目录
4 bootstrap.memory_lock: true #锁定内存
5 network.host: 10.0.0.51,127.0.0.1 #监听地址
6 http.port: 9200 #端口
7 discovery.seed_hosts: ["10.0.0.51"] #发现节点
8 cluster.initial_master_nodes: ["10.0.0.51"] #集群初始化节点
```

## 3.重启服务

```
1 systemctl restart elasticsearch.service
```

## 4.解决内存锁定失败

重启后查看日志发现提示内存锁定失败

```
1 [root@node-51 ~]# tail -f /var/log/elasticsearch/elasticsearch.log
2 [2020-12-17T19:34:38,132][ERROR][o.e.b.Bootstrap] [node-1] node
  validation exception
3 [1] bootstrap checks failed
4 [1]: memory locking requested for elasticsearch process but memory is not
  locked
```

官网解决方案：

```
1 https://www.elastic.co/guide/en/elasticsearch/reference/current/setting-
  system-settings.html#systemd
```

解决命令：

```
1 systemctl edit elasticsearch
2 [Service]
3 LimitMEMLOCK=infinity
4
5 systemctl daemon-reload
6 systemctl restart elasticsearch.service
```

# 第4章 Elasticsearch插件安装

## 1.elasticsearch-head介绍

```
1 官方地址：
2 https://github.com/mobz/elasticsearch-head
3
4 elasticsearch-head是一款用来管理Elasticsearch集群的第三方插件工具。
5 elasticsearch-Head插件在5.0版本之前可以直接以插件的形式直接安装，但是5.0以后安装方式
  发生了改变，需要nodejs环境支持，或者直接使用别人封装好的docker镜像，更推荐的是谷歌浏览器的
  插件。
```

## 2.elasticsearch-head的三种安装方式

- 1 | 1.npm安装方式
- 2 | 2.docker安装
- 3 | 3.google浏览器插件（推荐）

## 3.docker安装elasticsearch-head

```
1 | docker run -p 9100:9100 mobz/elasticsearch-head:7
```

## 3.npm安装elasticsearch-head

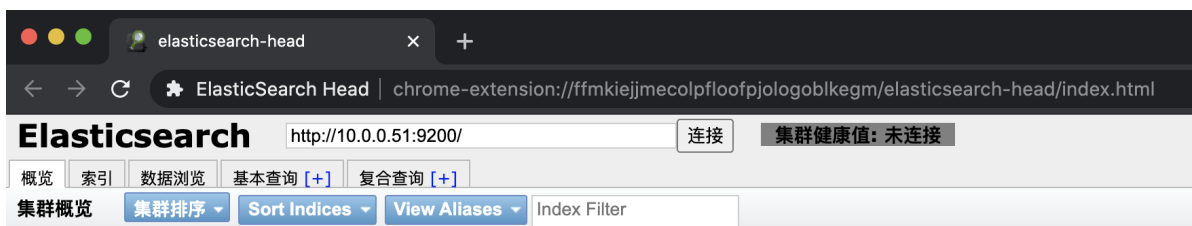
```
1 | cd /opt/  
2 | wget https://nodejs.org/dist/v12.13.0/node-v12.13.0-linux-x64.tar.xz  
3 | tar xf node-v12.13.0-linux-x64.tar.xz  
4 | mv node-v12.13.0-linux-x64 node  
5 | echo 'export PATH=$PATH:/opt/node/bin' >> /etc/profile  
6 | source /etc/profile  
7 | npm -v  
8 | node -v  
9 |  
10 | git clone git://github.com/mobz/elasticsearch-head.git  
11 | unzip elasticsearch-head-master.zip  
12 | cd elasticsearch-head-master  
13 |  
14 | npm install -g cnpm --registry=https://registry.npm.taobao.org  
15 | cnpm install  
16 | cnpm run start
```

修改Elasticsearch配置文件，添加如下参数并重启:

```
1 | http.cors.enabled: true  
2 | http.cors.allow-origin: "*"
```

## 4.es-head谷歌浏览器插件安装

更多工具-->拓展程序-->开发者模式-->选择解压缩后的插件目录



## 第5章 kibana安装

### 1.安装kibana

```
1 | rpm -ivh kibana-7.9.1-x86_64.rpm
```

### 2.配置kibana

```
1 [root@node-51 soft]# grep "^[a-z]" /etc/kibana/kibana.yml
2 server.port: 5601
3 server.host: "10.0.0.51"
4 elasticsearch.hosts: ["http://10.0.0.51:9200"]
5 kibana.index: ".kibana"
```

### 3.启动kibana

---

```
1 systemctl start kibana
```

### 4.检查测试

---

```
1 http://10.0.0.51:5601/
```



# Welcome to Elastic



## Let's get started

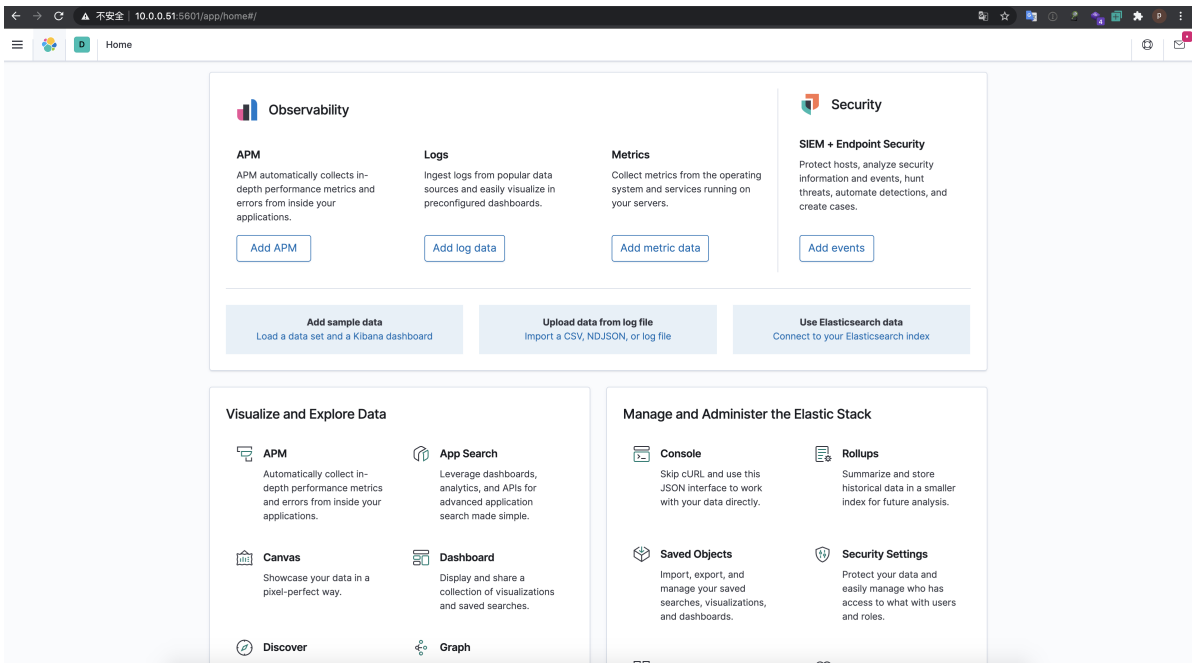
We noticed that you don't have any data in your cluster. You can try our sample data and dashboards or jump in with your own data.

[Try our sample data](#)

[Explore on my own](#)

To learn about how usage data helps us manage and improve our products and services, see our [Privacy Statement](#). To stop collection, [disable usage data here](#).





# 第6章 Elasticsearch插入命令

## 1.Elasticsearch数据格式

官网地址：

```
1 | https://www.elastic.co/guide/en/elasticsearch/reference/current/getting-started-index.html
```

和MySQL对比:

1	MySQL	Elasticsearch
2	库	索引 index
3	表	类型 _doc
4	字段	json数据的key
5	值	json数据的value
6	行	文档 doc

## 2.使用自定义的ID

curl命令操作:

```
1 | curl -XPUT 'http://10.0.0.51:9200/linux/_doc/1' -H 'Content-Type: application/json' -d '{
2 | {
3 |   "name": "zhang",
4 |   "age": "29"
5 | }'
```

kibana界面操作:

```
1 PUT linux/_doc/1
2 {
3   "name": "zhang",
4   "age": "29"
5 }
```

### 3.使用随机ID

```
1 POST linux/_doc/
2 {
3   "name": "zhang",
4   "age": "29",
5   "address": "BJ"
6 }
```

### 4.如何保证和mysql数据

```
1 mysql
2 id name age address job
3 1 zhang 27 BJ it
4 2 ya 22 SZ it
5
6 POST linux/_doc/
7 {
8   "id": "1",
9   "name": "zhang",
10  "age": "29",
11  "address": "BJ",
12  "job": "it"
13 }
14
15 POST linux/_doc/
16 {
17   "id": "2",
18   "name": "ya",
19   "age": "22",
20   "address": "SZ",
21   "job": "it"
22 }
```

## 第7章 Elasticsearch查询命令

### 1.创建测试语句

```
1 POST linux/_doc/
2 {
3   "name": "zhang3",
4   "age": "22",
5   "address": "SZ",
6   "job": "ops"
7 }
8
9 POST linux/_doc/
```

```
10 {
11   "name": "li4",
12   "age": "30",
13   "address": "BJ",
14   "job": "dev"
15 }
16
17 POST linux/_doc/
18 {
19   "name": "wang5",
20   "age": "24",
21   "address": "BJ",
22   "job": "dev"
23 }
24
25 POST linux/_doc/
26 {
27   "name": "zhao6",
28   "age": "35",
29   "address": "SZ",
30   "job": "devops"
31 }
32
33 POST linux/_doc/
34 {
35   "name": "sun7",
36   "age": "21",
37   "address": "BJ",
38   "job": "ops"
39 }
40
41 POST linux/_doc/
42 {
43   "name": "jack",
44   "age": "27",
45   "address": "BJ",
46   "job": "devops"
47 }
48
49 POST linux/_doc/
50 {
51   "name": "scott",
52   "age": "25",
53   "address": "SZ",
54   "job": "dev"
55 }
```

## 2.简单查询

---

```
1 GET linux/_search/
```

## 3.条件查询

---

```
1 GET linux/_search
2 {
```

```

3     "query": {
4       "term": {
5         "name": {
6           "value": "zhang3"
7         }
8       }
9     }
10  }
11
12  GET linux/_search
13  {
14    "query": {
15      "term": {
16        "job": {
17          "value": "ops"
18        }
19      }
20    }
21  }

```

## 4.多条件查询

```

1  GET linux/_search
2  {
3    "query": {
4      "bool": {
5        "must": [
6          {
7            "term": {
8              "job.keyword": "dev"
9            }
10           },
11           {
12             "term": {
13               "address.keyword": "BJ"
14             }
15           },
16           {
17             "range": {
18               "age.keyword": {
19                 "gt": "20",
20                 "lt": "30"
21               }
22             }
23           }
24         ]
25       }
26     }
27   }

```

# 第8章 Elasticsearch更新命令

## 1.自定义的ID更新

```
1 PUT linux/info/1
2 {
3   "name": "zhang",
4   "age": 30,
5   "job": "it",
6   "id": 1
7 }
```

## 2.随机ID更新

创建测试数据

```
1 PUT linux/_doc/1
2 {
3   "name": "zhang",
4   "age": "30",
5   "job": "it",
6   "id": 2
7 }
```

先根据自定义的id字段查出数据的随机ID

```
1 GET linux/_search/
2 {
3   "query": {
4     "term": {
5       "id": {
6         "value": "2"
7       }
8     }
9   }
10 }
```

取到随机ID后更改数据

```
1 PUT linux/_doc/CVDdknIBq3aq7mPQaoWw
2 {
3   "name": "tony",
4   "age": 30,
5   "job": "it",
6   "id": 2
7 }
```

## 第9章 Elasticsearch集群概念介绍

### 1.Elasticsearch集群特点

- 1 对运维友好：不需要太多java的知识也可以很方便的维护整个集群。
- 2 搭建方便：搭建副本非常简单，只需要将新节点加入已有集群即可，会自动同步数据。
- 3 自动故障转移：当节点出现故障时，会自动故障转移，将有数据复制到其他正常的节点。

### 2.数据分片

- 1 | 主分片：实际存储的数据,负责读写,粗框的是主分片
- 2 | 副本分片：主分片的副本,提供读,同步主分片,细框的是副本分片

## 3.副本

- 1 | 主分片的备份,副本数量可以自定义

## 4.默认分片和副本规则

- 1 | 7.x版本之前默认规则：1副本，5分片
- 2 | 7.x版本之后默认规则：1副本，1分片

## 5.节点类型

- 1 | 主节点：负责调度数据分配到哪个节点
- 2 | 数据节点：实际负责处理数据的节点
- 3 | 默认：主节点也是工作节点

## 6.集群健康状态

- 1 | 绿色：所有数据都完整，且副本数满足
- 2 | 黄色：所有数据都完整，但是副本数不满足
- 3 | 红色：一个或多个索引数据不完整

# 第10章: Elasticsearch集群部署

## 1.部署集群前注意事项

- 1 | 最好是使用干净的环境部署集群，如果以前有单节点的数据，最好备份出来，然后再清空集群数据。

## 2.新节点安装java

- 1 | 7.x版本之后不需要单独的安装JDK，软件包自带了JDK

## 3.新节点安装Elasticsearch

- 1 | `rpm -ivh elasticsearch-7.9.1-x86_64.rpm`

## 4.配置内存锁定

- 1 | `systemctl edit elasticsearch.service`
- 2 | `[Service]`
- 3 | `LimitMEMLOCK=infinity`

## 5.集群配置文件

## 5.1 配置文件解释

```
1 cluster.name: oldboy_linux      #集群名称
2 node.name: node-1              #节点名称
3 path.data: /var/lib/elasticsearch #数据目录
4 path.logs: /var/log/elasticsearch #日志目录
5 bootstrap.memory_lock: true     #设置内存锁定
6 network.host: 127.0.0.1,10.0.0.51 #本地监听地址
7 http.port: 9200                 #本地端口
8 discovery.seed_hosts: ["10.0.0.51","10.0.0.52"] #集群节点互相发现的地址，不需要把所有节点IP都写上。
9 cluster.initial_master_nodes: ["10.0.0.51"]     #集群初始化节点，只有创建集群的第一次有用，集群创建后参数失效。
```

## 5.2 node1配置文件:

```
1 cat > /etc/elasticsearch/elasticsearch.yml <<EOF
2 cluster.name: oldboy_linux
3 node.name: node-1
4 path.data: /var/lib/elasticsearch
5 path.logs: /var/log/elasticsearch
6 bootstrap.memory_lock: true
7 network.host: 127.0.0.1,10.0.0.51
8 http.port: 9200
9 discovery.seed_hosts: ["10.0.0.51","10.0.0.52"]
10 cluster.initial_master_nodes: ["10.0.0.51"]
11 EOF
```

## 5.3 node2配置文件:

```
1 cat> /etc/elasticsearch/elasticsearch.yml <<EOF
2 cluster.name: oldboy_linux
3 node.name: node-2
4 path.data: /var/lib/elasticsearch
5 path.logs: /var/log/elasticsearch
6 bootstrap.memory_lock: true
7 network.host: 127.0.0.1,10.0.0.52
8 http.port: 9200
9 discovery.seed_hosts: ["10.0.0.51","10.0.0.52"]
10 cluster.initial_master_nodes: ["10.0.0.51"]
11 EOF
```

## 6.启动

注意：如果以前单节点有数据，那么先停止运行，然后清空数据

```
1 systemctl stop elasticsearch.service
2 rm -rf /var/lib/elasticsearch/*
```

重启命令:

```
1 systemctl daemon-reload
2 systemctl restart elasticsearch
```

## 7.查看日志

```
1 | tail -f /var/log/elasticsearch/oldboy_linux.log
```

## 8.检查集群

```
1 | ES-head查看是否有2个节点
```



## 9.集群注意事项

1. 插入和读取数据在任意节点都可以执行,效果一样
2. es-head可以连接集群内任一服务
3. 主节点负责读写  
如果主分片所在的节点坏掉了,副本分片会升为主分片
4. 主节点负责调度  
如果主节点坏掉了,数据节点会自动升为主节点
5. 通讯端口  
默认会有2个通讯端口: 9200和9300  
9300并没有在配置文件里配置过  
如果开启了防火墙并且没有放开9300端口,那么集群通讯就会失败

# 第11章 Elasticsearch集群扩容

## 1.安装java

```
1 | 7.0版本之后不需要单独安装JDK
```

## 2.安装ES

```
1 | rpm -ivh elasticsearch-7.9.1-x86_64.rpm
```

## 3.配置内存锁定



```
1 | systemctl edit elasticsearch.service
2 | [Service]
3 | LimitMEMLOCK=infinity
```

## 4.node3集群配置文件

```
1 | cat> /etc/elasticsearch/elasticsearch.yml <<EOF
2 | cluster.name: oldboy_linux
3 | node.name: node-3
4 | path.data: /var/lib/elasticsearch
5 | path.logs: /var/log/elasticsearch
6 | bootstrap.memory_lock: true
7 | network.host: 127.0.0.1,10.0.0.53
8 | http.port: 9200
9 | discovery.seed_hosts: ["10.0.0.51","10.0.0.53"]
10 | cluster.initial_master_nodes: ["10.0.0.51"]
11 | EOF
```

## 5.添加节点注意

```
1 | 对于新添加的节点来说：
2 | 只需要直到集群内任意一个节点的IP和他自己本身的IP即可
3 | discovery.seed_hosts: ["10.0.0.51","10.0.0.53"]
4 |
5 | 对于以前的节点来说：
6 | 什么都不需要更改
```

## 6.数据分片颜色解释

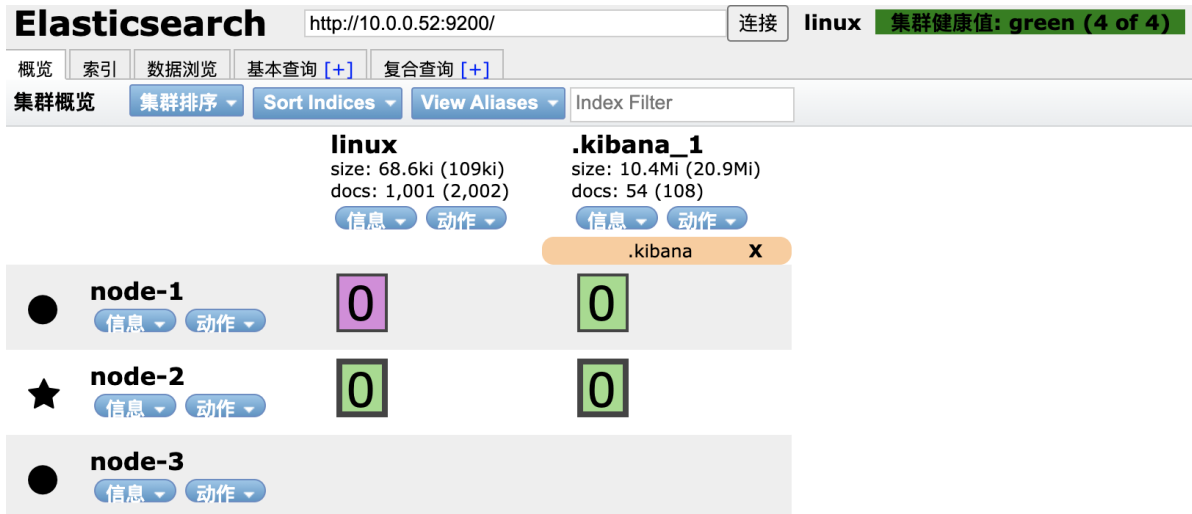
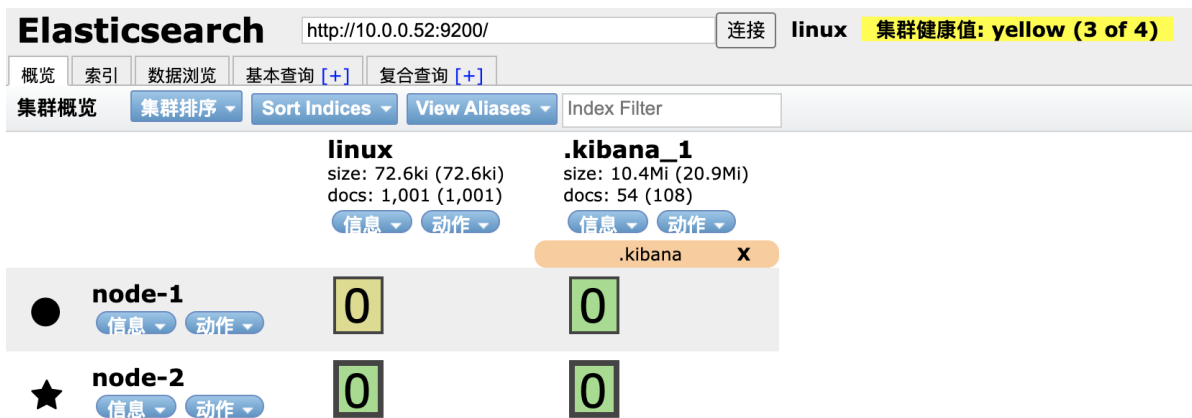
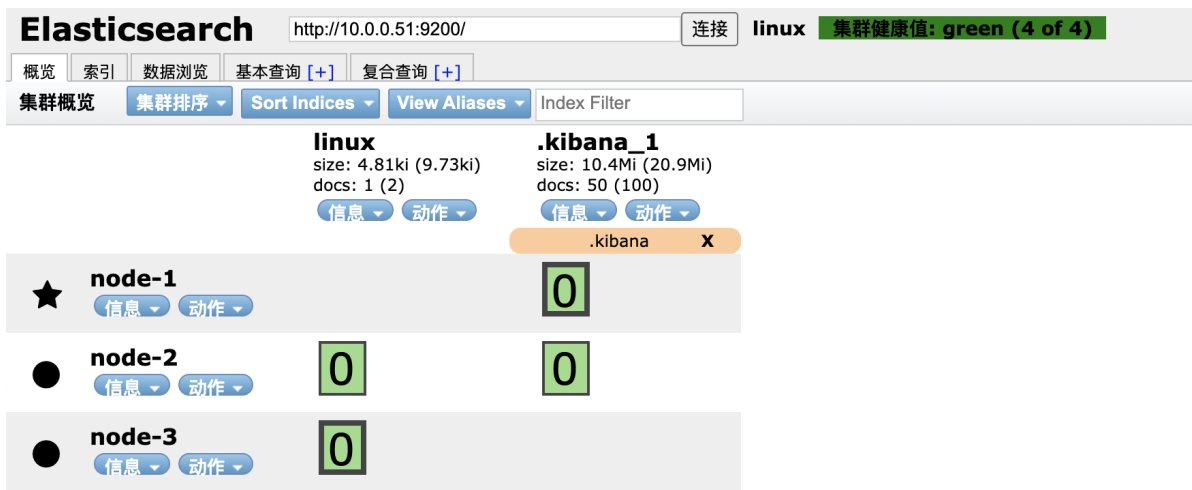
```
1 | 紫色：正在迁移
2 | 黄色：正在复制
3 | 绿色：正常
```

## 7.集群故障转移实验

```
1 | 1.停掉主节点，观察集群是否正常
2 | 2.停掉主节点，是否还会选举出新的主节点
3 | 3.停掉主节点，数据分片的分布会不会发生变化，分片状态会不会发生变化
4 | 4.停掉主节点，然后在持续的写入数据，等节点恢复之后，会如何处理落后的数据
5 | 5.3个节点的Elasticsearch集群，极限情况下最多允许坏几台？
6 | 6.主节点故障，集群健康状态发生什么变化？
```

结论：

```
1 | 1.如果主节点坏掉了，会从活着的数据节点中选出一台新的主节点
2 | 2.如果主分片坏掉了，副本分片会升级为主分片
3 | 3.如果副本数不满足，会尝试在其他的节点上重新复制一份数据
4 | 4.修复上线只需要正常启动故障的节点即会自动加入到集群里，并且自动同步数据
5 | 5.7.x版本之后则必须至少2个节点存活集群才能正常工作
```



## 第12章 Elasticsearch集群维护

### 1.自定义副本数和索引数参数注意事项

- 1 索引一旦建立完成,分片数就不可以修改了
- 2 但是副本数可以随时修改

### 2.创建索引的时候就自定义副本和分片

```

1 PUT /linux2/
2 {
3   "settings": {
4     "number_of_shards": 3,
5     "number_of_replicas": 0
6   }
7 }

```

### 3.修改单个索引的副本数

```

1 PUT /linux2/_settings/
2 {
3   "settings": {
4     "number_of_replicas": 2
5   }
6 }

```

### 4.修改所有的索引的副本数

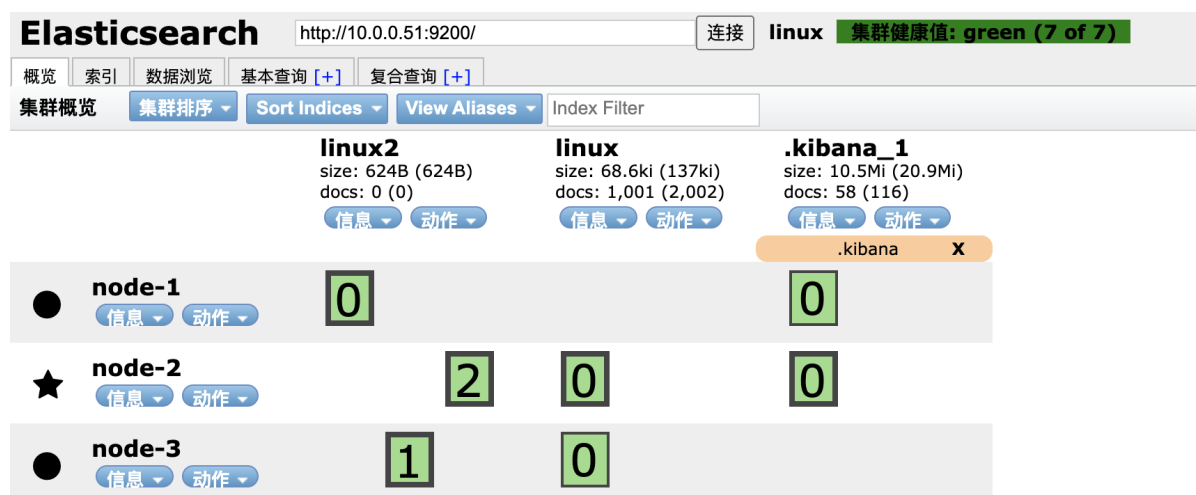
```

1 PUT /_all/_settings/
2 {
3   "settings": {
4     "number_of_replicas": 0
5   }
6 }

```

### 5.工作如何设置

- 1 2个节点：默认就可以
- 2 3个节点：重要的数据,2副本 不重要的默认
- 3 日志收集：1副本3分片



Elasticsearch
http://10.0.0.51:9200/
连接
linux
集群健康值: red (4 of 7)

概览
索引
数据浏览
基本查询 [+]
复合查询 [+]

集群概览
集群排序
Sort Indices
View Aliases
Index Filter

	linux2	linux	.kibana_1
size: 416B (416B)	size: 68.6ki (68.6ki)	size: 10.5Mi (10.5Mi)	
docs: 0 (0)	docs: 1,001 (1,001)	docs: 60 (60)	
信息	信息	信息	
动作	动作	动作	
			.kibana X
Unassigned	2	0	0
node-1	0		0
node-3	1	0	

# 第13章 Ela监控

## 1.监控注意

- 1

1.不能只监控集群状态
- 2

2.监控节点数
- 3

3.监控集群状态
- 4

4.两者任意一个发生改变了都报警

## 2.监控命令

- 1

GET \_cat/nodes
- 2

GET \_cat/health
- 3

GET \_cat/master
- 4

GET \_cat/indices
- 5

GET \_cat/shards
- 6

GET \_cat/shards/linux

查看集群健康状态

- 1

curl -s 127.0.0.1:9200/\_cat/health|grep "green"|wc -l

查看节点个数

- 1

curl -s 127.0.0.1:9200/\_cat/nodes|wc -l

## 3.kibana开启监控

- 1

点击kibana面板的监控按钮

## 4.kibana关闭监控

```
1 GET /_cluster/settings
2 PUT /_cluster/settings
3 {
4   "persistent" : {
5     "xpack" : {
6       "monitoring" : {
7         "collection" : {
8           "enabled" : "false"
9         }
10      }
11    }
12  }
13 }
```

## 第14章 中文分词器

### 1.未分词的情况

#### 1.1 插入测试数据

```
1 POST /news/_doc/1
2 {"content": "美国留给伊拉克的是个烂摊子吗"}
3
4 POST /news/_doc/2
5 {"content": "公安部：各地校车将享最高路权"}
6
7 POST /news/_doc/3
8 {"content": "中韩渔警冲突调查：韩警平均每天扣1艘中国渔船"}
9
10 POST /news/_doc/4
11 {"content": "中国驻洛杉矶领事馆遭亚裔男子枪击 嫌犯已自首"}
```

#### 1.2 查询测试

```
1 POST /news/_search
2 {
3   "query" : { "match" : { "content" : "中国" }},
4   "highlight" : {
5     "pre_tags" : [<tag1>, "<tag2>"],
6     "post_tags" : [</tag1>, "</tag2>"],
7     "fields" : {
8       "content" : {}
9     }
10  }
11 }
```

#### 1.3 结论

1 未配置中文分词器时查询中文会将词拆分成一个一个的汉字。

### 2.中文分词配置

## 2.1 前提条件

- 1 所有的ES节点都需要安装
- 2 所有的ES都需要重启才能生效
- 3 中文分词器的版本号要和ES版本号对应
- 4 <https://github.com/medcl/elasticsearch-analysis-ik>

## 2.2 配置中文分词器

在线安装

- ```
1 /usr/share/elasticsearch/bin/elasticsearch-plugin install
  https://github.com/medcl/elasticsearch-analysis-
  ik/releases/download/v7.9.1/elasticsearch-analysis-ik-7.9.1.zip
```

离线本地文件安装

- ```
1 /usr/share/elasticsearch/bin/elasticsearch-plugin install
  file:///opt/elasticsearch-analysis-ik-7.9.1.zip
```

## 2.3 重启所有ES节点

- ```
1 systemctl restart elasticsearch.service
```

## 2.4 创建索引

- ```
1 PUT /news2
```

## 2.5 创建模板

- ```
1 POST /news2/_doc/_mapping?include_type_name=true
2 {
3   "properties": {
4     "content": {
5       "type": "text",
6       "analyzer": "ik_max_word",
7       "search_analyzer": "ik_smart"
8     }
9   }
10 }
```

## 2.6 插入测试数据

```
1 POST /news2/_doc/1
2 {"content": "美国留给伊拉克的是个烂摊子吗"}
3
4 POST /news2/_doc/2
5 {"content": "公安部：各地校车将享最高路权"}
6
7 POST /news2/_doc/3
8 {"content": "中韩渔警冲突调查：韩警平均每天扣1艘中国渔船"}
9
10 POST /news2/_doc/4
11 {"content": "中国驻洛杉矶领事馆遭亚裔男子枪击 嫌犯已自首"}
```

## 2.7 再次查询数据发现已经能识别中文了

```
1 POST /news2/_search
2 {
3   "query" : { "match" : { "content" : "中国" }},
4   "highlight" : {
5     "pre_tags" : ["<tag1>", "<tag2>"],
6     "post_tags" : ["</tag1>", "</tag2>"],
7     "fields" : {
8       "content" : {}
9     }
10  }
11 }
```

## 3.热更新中文分词库

### 3.1 安装nginx

```
1 yum install nginx -y
```

### 3.2 编写字典文件

```
1 cat >>/usr/share/nginx/html/my_dic.txt<<EOF
2 北京
3 张亚
4 武汉
5 中国
6 深圳
7 EOF
```

### 3.3 重启并测试

```
1 nginx -t
2 systemctl restart nginx
3 curl 127.0.0.1/my_dic.txt
```

### 3.4 配置es的中文分词器插件

```
1 cat >/etc/elasticsearch/analysis-ik/IKAnalyzer.cfg.xml<<'EOF'
2 <?xml version="1.0" encoding="UTF-8"?>
3 <!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
4 <properties>
5     <comment>IK Analyzer 扩展配置</comment>
6     <!--用户可以在这里配置自己的扩展字典 -->
7     <entry key="ext_dict"></entry>
8     <!--用户可以在这里配置自己的扩展停止词字典-->
9     <entry key="ext_stopwords"></entry>
10    <!--用户可以在这里配置远程扩展字典 -->
11    <entry key="remote_ext_dict">http://10.0.0.51/my_dic.txt</entry>
12    <!--用户可以在这里配置远程扩展停止词字典-->
13    <!-- <entry key="remote_ext_stopwords">words_location</entry> -->
14 </properties>
15 EOF
```

### 3.5 将修改好的IK配置文件复制到其他所有ES节点

```
1 cd /etc/elasticsearch/analysis-ik/
2 scp IKAnalyzer.cfg.xml 10.0.0.52:/etc/elasticsearch/analysis-ik/
3 scp IKAnalyzer.cfg.xml 10.0.0.53:/etc/elasticsearch/analysis-ik/
```

### 3.6 重启所有的ES节点

```
1 systemctl restart elasticsearch.service
```

### 3.7 查看日志里字典的词有没有加载出来

```
1 [2020-12-18T10:27:08,126][INFO ][o.w.a.d.Dictionary      ] [node-1] start to
  reload ik dict.
2 [2020-12-18T10:27:08,127][INFO ][o.w.a.d.Dictionary      ] [node-1] try load
  config from /etc/elasticsearch/analysis-ik/IKAnalyzer.cfg.xml
3 [2020-12-18T10:27:08,538][INFO ][o.w.a.d.Dictionary      ] [node-1] [Dict
  Loading] http://10.0.0.51/my_dic.txt
4 [2020-12-18T10:27:08,540][INFO ][o.w.a.d.Dictionary      ] [node-1] 北京
5 [2020-12-18T10:27:08,541][INFO ][o.w.a.d.Dictionary      ] [node-1] 张亚
6 [2020-12-18T10:27:08,541][INFO ][o.w.a.d.Dictionary      ] [node-1] 武汉
7 [2020-12-18T10:27:08,541][INFO ][o.w.a.d.Dictionary      ] [node-1] 中国
8 [2020-12-18T10:27:08,541][INFO ][o.w.a.d.Dictionary      ] [node-1] 深圳
9 [2020-12-18T10:27:08,541][INFO ][o.w.a.d.Dictionary      ] [node-1] reload ik
  dict finished.
```

### 3.8 打开es日志，然后更新字典内容，查看日志里会不会自动加载

```
1 echo "老男孩教育" >> /usr/share/nginx/html/my_dic.txt
```



## 3.9 搜索测试验证结果

```
1 POST /news2/_doc/7
2 {
3   "content": "学Linux来老男孩教育"
4 }
5
6
7 POST /news2/_search
8 {
9   "query" : { "match" : { "content" : "老男孩教育" }},
10  "highlight" : {
11    "pre_tags" : [<tag1>, "<tag2>"],
12    "post_tags" : [</tag1>, "</tag2>"],
13    "fields" : {
14      "content" : {}
15    }
16  }
17 }
```

## 3.10 电商上架新产品流程

- 1 先把新上架的商品的关键词更新到词典里
- 2 查看ES日志，确认新词被动态更新了
- 3 自己编写一个测试索引，插入测试数据，然后查看搜索结果
- 4 确认没有问题之后，在让开发插入新商品的数据
- 5 测试

# 第15章 备份恢复

## 1.使用官方的快照snap功能备份恢复

### 1.1 前提条件

官方地址:

- 1 <https://www.elastic.co/guide/en/elasticsearch/reference/7.9/snapshot-restore.html>

前提条件:

- 1 如果是Elasticsearch集群想使用快照功能，则存储快照的目录必须是共享存储，并且所有节点都需要挂载这个目录。

配置NFS命令:

```
1 #服务端配置
2 yum install nfs-utils -y
3 cat > /etc/exports << 'EOF'
4 /data/backup 10.0.0.0/24(rw,sync,all_squash,anonuid=997,anongid=995)
5 EOF
6 systemctl restart nfs
7 showmount -e 10.0.0.51
```

```
8 | mkdir /data/backup -p
9 |
10 | #客户端配置
11 | yum install nfs-utils -y
12 | mkdir /data/backup -p
13 | mount -t nfs 10.0.0.51:/data/backup /data/backup
14 | df -h
```

## 1.2 创建目录

```
1 | mkdir /data/backup -p
2 | chown -R elasticsearch:elasticsearch /data/backup/
```

## 1.3 所有节点修改Elasticsearch配置文件，添加参数

```
1 | path.repo: ["/data/backup"]
```

## 1.4 重启ES

```
1 | systemctl restart elasticsearch
```

## 1.5 注册快照

```
1 | PUT /_snapshot/my_fs_backup
2 | {
3 |     "type": "fs",
4 |     "settings": {
5 |         "location": "/data/backup/my_fs_backup_location",
6 |         "compress": true
7 |     }
8 | }
```

## 1.6 查看快照

```
1 | GET /_snapshot/my_fs_backup
```

## 1.7 创建第一个快照

```
1 | PUT /_snapshot/my_fs_backup/snapshot_1?wait_for_completion=true
```

## 1.8 创建指定索引的快照

```
1 | PUT /_snapshot/my_fs_backup/snapshot_2?wait_for_completion=true
2 | {
3 |     "indices": "news,news2",
4 |     "ignore_unavailable": true,
5 |     "include_global_state": false
6 | }
```

## 1.9 查询快照信息

```
1 GET /_snapshot/my_fs_backup/snapshot_1
2 GET /_snapshot/my_fs_backup/snapshot_2
```

## 1.10 查看正在运行的快照

```
1 GET /_snapshot/my_fs_backup/_current
```

## 1.11 删除快照

```
1 DELETE /_snapshot/my_fs_backup/snapshot_2
```

## 1.12 删除存储库

```
1 DELETE /_snapshot/my_fs_backup
```

## 1.13 全部还原

```
1 POST /_snapshot/my_fs_backup/snapshot_1/_restore
```

## 1.14 还原部分

```
1 POST /_snapshot/my_fs_backup/snapshot_1/_restore
2 {
3   "indices": "news,news2",
4   "ignore_unavailable": true,
5   "include_global_state": true,
6   "rename_pattern": "news_(.+)",
7   "rename_replacement": "restored_news_$1"
8 }
```

## 1.15 恢复的同时更改索引配置

```
1 POST /_snapshot/my_fs_backup/snapshot_1/_restore
2 {
3   "indices": "index_1",
4   "index_settings": {
5     "index.number_of_replicas": 0
6   },
7   "ignore_index_settings": [
8     "index.refresh_interval"
9   ]
10 }
```

## 1.16 以日期命名快照

```
1 PUT /_snapshot/my_fs_backup/%3Csnapshot-%7Bnow%2Fd%7D%3E
2 GET /_snapshot/my_fs_backup/_all
```

## 2.使用第三方工具elasticdump备份恢复

### 2.1 前提条件

需要node环境

```
1 npm -v
2 node -v
```

### 2.2 nodejs安装

```
1 wget https://nodejs.org/dist/v10.16.3/node-v10.16.3-linux-x64.tar.xz
2 tar xf node-v10.16.3-linux-x64.tar.xz -C /opt/
3 cd /opt/
4 ln -s node-v10.16.3-linux-x64 node
5 echo 'export PATH=/opt/node/bin:$PATH' >> /etc/profile
6 source /etc/profile
7 npm -v
8 node -v
```

### 2.3 指定使用国内淘宝npm源

```
1 npm install -g cnpm --registry=https://registry.npm.taobao.org
```

### 2.4 安装es-dump

```
1 cnpm install elasticdump -g
```

### 2.5 备份

备份成可读的json格式

```
1 elasticdump \
2 --input=http://10.0.0.51:9200/news2 \
3 --output=/data/news2.json \
4 --type=data
```

备份成压缩格式

```
1 elasticdump \
2 --input=http://10.0.0.51:9200/news2 \
3 --output=$|gzip > /data/news2.json.gz
```

备份分词器/mapping/数据一条龙服务

```
1 elasticdump \  
2   --input=http://10.0.0.51:9200/news2 \  
3   --output=/data/news2_mapping.json \  
4   --type=mapping\  
5 elasticdump \  
6   --input=http://10.0.0.51:9200/news2 \  
7   --output=/data/news2.json \  
8   --type=data
```

## 2.6 恢复

只恢复数据

```
1 elasticdump \  
2   --input=/data/news2.json \  
3   --output=http://10.0.0.51:9200/news2
```

恢复所有数据包含分词器/mapping一条龙

```
1 elasticdump \  
2   --input=/data/news2_mapping.json \  
3   --output=http://10.0.0.51:9200/news2 \  
4   --type=mapping\  
5 elasticdump \  
6   --input=/data/news2.json \  
7   --output=http://10.0.0.51:9200/news2 \  
8   --type=data
```

## 2.7 批量备份

```
1 curl -s 10.0.0.52:9200/_cat/indices|awk '{print $3}'|grep -v "^\."
```

## 2.8 注意事项

1. 如果恢复的时候数据冲突了，会被覆盖掉
2. 如果已经存在备份文件里没有的数据，会保留下来

## 2.9 带密码认证的导出

```
1 --input=http://name:password@production.es.com:9200/my_index
```

# 第16章 安全认证

## 1. 官方地址

```
1 https://www.elastic.co/guide/en/elasticsearch/reference/current/configuring-security.html
```

## 2. 生成证书和密钥

```
1 /usr/share/elasticsearch/bin/elasticsearch-certutil ca
2 /usr/share/elasticsearch/bin/elasticsearch-certutil cert --ca elastic-stack-
  ca.p12
```

### 3.复制证书到合适的位置并复制到集群所有节点

```
1 mkdir /etc/elasticsearch/certs
2 cp /usr/share/elasticsearch/*.p12 /etc/elasticsearch/certs/
3 chown -R elasticsearch:elasticsearch /etc/elasticsearch/certs/
4 scp -r /etc/elasticsearch/certs 10.0.0.52:/etc/elasticsearch/
5 scp -r /etc/elasticsearch/certs 10.0.0.53:/etc/elasticsearch/
```

### 3.修改配置文件开启安全功能

```
1 xpack.security.enabled: true
2 xpack.security.transport.ssl.enabled: true
3 xpack.security.transport.ssl.verification_mode: certificate
4 xpack.security.transport.ssl.keystore.path: certs/elastic-stack-ca.p12
5 xpack.security.transport.ssl.truststore.path: certs/elastic-stack-ca.p12
```

### 3.重启所有节点

```
1 systemctl restart elasticsearch
```

### 4.配置用户密码

```
1 /usr/share/elasticsearch/bin/elasticsearch-setup-passwords interactive
```

内置账户角色说明:

- 1 **elastic**: 拥有superuser角色,是内置的超级用户。
- 2 **kibana**: 拥有kibana\_system角色,用户kibana用来连接elasticsearch并与之通信。Kibana服务器以该用户身份提交请求以访问集群监视API和.kibana索引。不能访问index。
- 3 **logstash\_system**账号: 拥有logstash\_system角色。用户Logstash在Elasticsearch中存储监控信息时使用。
- 4 **beats\_system**账号: 拥有beats\_system角色。用户Beats在Elasticsearch中存储监控信息时使用。

详细权限的官网说明:

```
1 https://www.elastic.co/guide/en/elasticsearch/reference/7.9/security-
  privileges.html
```

### 5.kibana配置密码认证

```
1 vim /etc/kibana/kibana.yml
2 elasticsearch.username: "kibana_system"
3 elasticsearch.password: "elastic"
```

修改好配置后记得重启:

```
1 | systemctl restart kibana
```

## 6.访问测试



Welcome to Elastic

You have logged out of Elastic.

**Username**

elastic

**Password**

••••••

Log in

## 7.创建角色

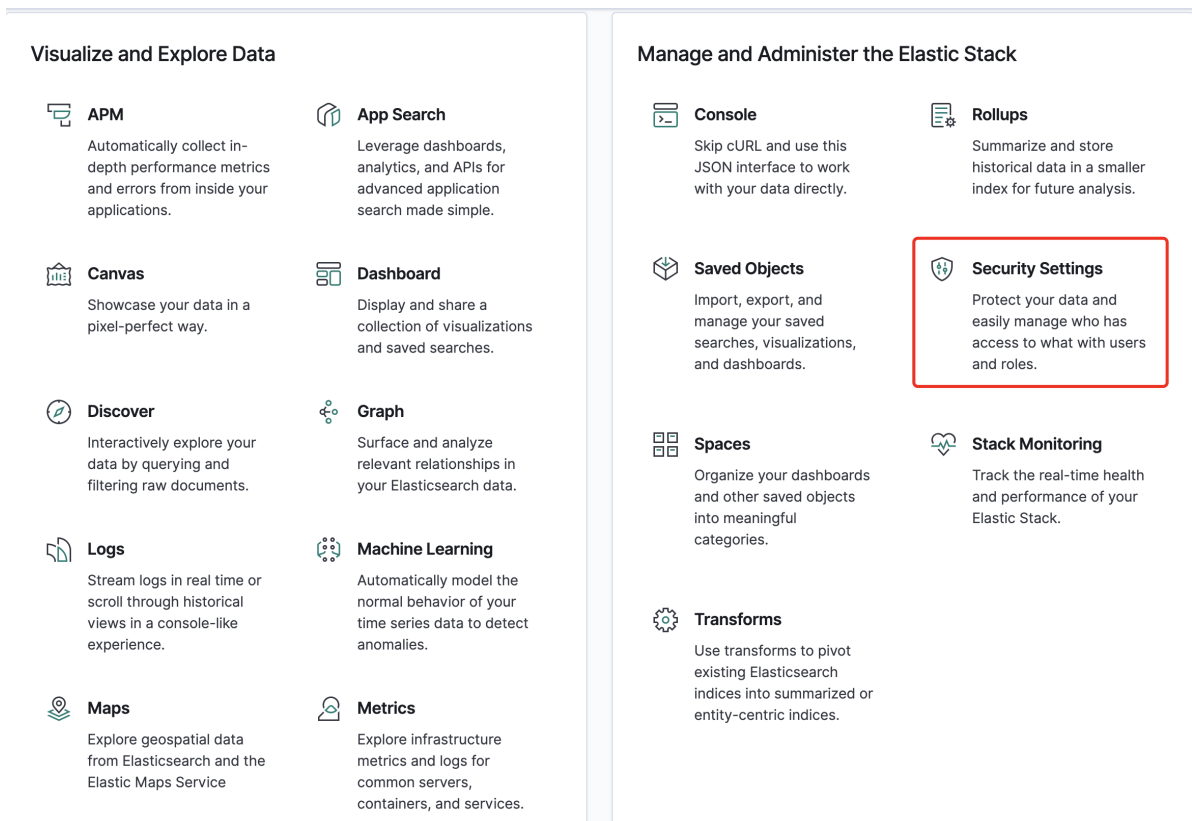


image-20201223193554212

## 8.创建用户

## 9.

## x.注意事项

1. 一定要先配置证书认证，再配置密码，不然就会报错。
2. 默认创建的证书是root只读权限，需要设置为elasticsearch可以读取的权限，不然启动报错
3. 创建账号密码命令只需要在master节点配置即可
4. 创建账号密码命令只能运行一次，再次运行就会报错
5. 初始化的账号密码仅仅是作为组件之间传输信息使用，并不是给用户使用的。
6. 如果是需要给用户分配权限，需要在kibana里以elastic用户登录，然后在创建新用户和新角色。

## 第17章 ES优化

### 1.官方参考

- 1 <https://www.elastic.co/guide/en/elasticsearch/reference/current/system-config.html>

### 2.优化建议



- 1 1.内存
- 2 1.系统建议预留一半
- 3 2.每个ES节点不要超过32G
- 4 3.关闭swap分区
- 5 4.配置文件打开内存锁定参数
- 6 5.升级SSD硬盘
- 7 6.升级大版本