主讲老师： Fox

课前须知：

学习本节课之前，需要先掌握ES高级查询语法Query DSL实战 ElasticSearch搜索技术深入与聚合查询实战

```
文档：5 电商项目ElasticSearch高性能搜索实?..
链接：http://note.youdao.com/noteshare?id=48251867771266d6b9d2d4d1b5b14b27&sub=95D841A7B8DC40718C6DDDB35199F055
```

# 1.ElasticSearch回顾

ElasticSearch（简称ES）是一个分布式、RESTful 风格的搜索和数据分析引擎，是用Java开发并且是当前最流行的开源的企业级搜索引擎，能够达到近实时搜索，稳定，可靠，快速，安装使用方便。

客户端支持Java、.NET（C#）、PHP、Python、Ruby等多种语言。

**官方网站:** https://www.elastic.co/

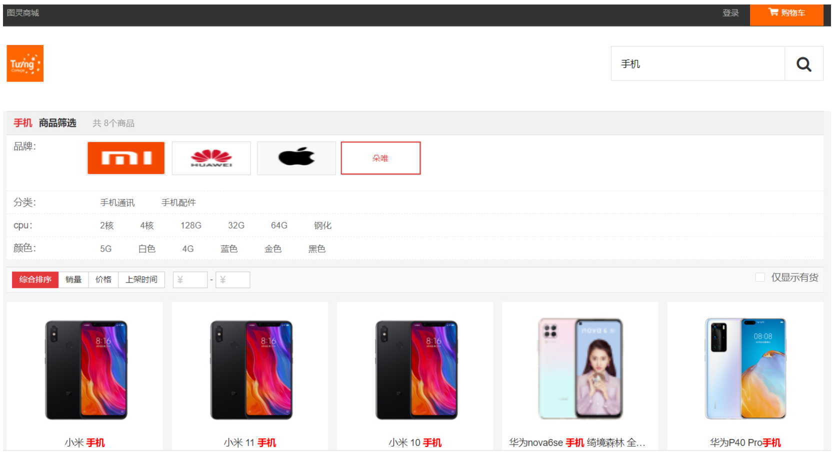
**下载地址：** https://www.elastic.co/cn/downloads/past-releases#elasticsearch

**应用场景**

- 站内搜索
- 日志管理与分析
- 大数据分析
- 应用性能监控
- 机器学习

## 1.1 业务场景——图灵商城商品搜索

根据关键字查询、根据品牌查询、商品类别、商品属性信息、价格区间、是否有库存、排序(销量、价格、上架时间等)



## 1.2 文档建模

商品json文档

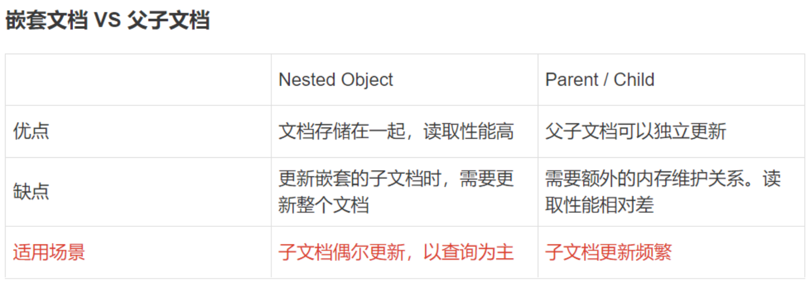
```
{
 "id": "26",
 "name": "小米 11 手机",
 "keywords": "小米手机",
 "subTitle": "AI智慧全面屏 6GB +64GB 亮黑色 全网通版 移动联通电信4G手机 双卡双待 双卡双待",
 "price": "3999",
 "promotionPrice": "2999",
 "originalPrice": "5999",
 "pic": "http://macro-oss.oss-cn-shenzhen.aliyuncs.com/mall/images/20180615/xiaomi.jpg",
  "sale": 999,
  "hasStock": true,
  "salecount":999,
  "putawayDate":"2021-04-01",
  "brandId": 6,
  "brandName": "小米",
  "brandImg": "http://macro-oss.oss-cn-shenzhen.aliyuncs.com/mall/images/20190129/1e34aef2a409119018a4c6258e39ecfb_222_222.png",
  "categoryId": 19,
  "categoryName": "手机通讯",
  "attrs": [
  {
  "attrId": 1,
  "attrName": "cpu",
  "attrValue": "2核"
  },
  {
  "attrId": 2,
  "attrName": "颜色",
  "attrValue": "黑色"
  }
  ]
 }

 {
  "id": "30",
  "name": "HLA海澜之家简约动物印花短袖T恤",
  "keywords": "海澜之家衣服",
  "subTitle": "HLA海澜之家短袖T恤",
  "price": "199",
  "promotionPrice": "99",
  "originalPrice": "299",
  "pic": "http://macro-oss.oss-cn-shenzhen.aliyuncs.com/mall/images/20180615/5ad83a4fN6ff67ecd.jpg!cc_350x449.jpg",
  "sale": 999,
  "hasStock": true,
  "salecount":19,
  "putawayDate":"2021-04-05",
  "brandId": 50,
  "brandName": "海澜之家",
  "brandImg": "http://macro-oss.oss-cn-shenzhen.aliyuncs.com/mall/images/20190129/99d3279f1029d32b929343b09d3c72de_222_222.jpg",
  "categoryId": 8,
  "categoryName": "T恤",
  "attrs": [
  {
  "attrId": 3,
  "attrName": "尺寸",
  "attrValue": "M"
  },
  {
  "attrId": 4,
  "attrName": "颜色",
  "attrValue": "黑色"
  }
  ]
 }
```

建模分析：

- name，keywords，subTitle 需要使用中文分词器
- categoryName，brandName 类型可以为keyword
- 不同的商品其属性也不同，属性和商品之间存在关联关系。商品属性attrs不会频繁更新，可以选择使用nested类型

思考：Elasticsearch中如何处理关联关系？



**嵌套文档 VS 父子文档**

| | Nested Object | Parent / Child |
|---|---|---|
| 优点 | 文档存储在一起，读取性能高 | 父子文档可以独立更新 |
| 缺点 | 更新嵌套的子文档时，需要更新整个文档 | 需要额外的内存维护关系。读取性能相对差 |
| 适用场景 | 子文档偶尔更新，以查询为主 | 子文档更新频繁 |

定义mapping，创建索引库

```
PUT product_db
{
 "mappings": {
 "properties": {
 "id": {
 "type": "long"
 },
 "name": {
 "type": "text",
  "analyzer": "ik_max_word"
  },
  "keywords": {
  "type": "text",
  "analyzer": "ik_max_word"
  },
  "subTitle": {
  "type": "text",
  "analyzer": "ik_max_word"
  },
  "salecount":{
  "type": "long"
 },
 "putawayDate":{
 "type": "date"
 },
 "price": {
 "type": "double"
 },

 "promotionPrice": {
 "type": "keyword"
 },
 "originalPrice": {
 "type": "keyword"
 },
 "pic": {
 "type": "keyword"
 },
 "sale": {
 "type": "long"
 },
 "hasStock": {
 "type": "boolean"
 },
 "brandId": {
 "type": "long"
 },
 "brandName": {
 "type": "keyword"
 },
 "brandImg": {
 "type": "keyword"
 },
 "categoryId": {
 "type": "long"
 },
 "categoryName": {
 "type": "keyword"
 },
 "attrs": {
 "type": "nested",
 "properties": {
 "attrId": {
 "type": "long"
 },
 "attrName": {
 "type": "keyword"
 },
 "attrValue": {
 "type": "keyword"
 }
 }
 }
 }
 }
}

```

## 1.3 索引文档

可以利用canal进行数据同步

```
PUT /product_db/_doc/1
```

```
{
 "id": "26",
 "name": "小米 11 手机",
 "keywords": "小米手机",
 "subTitle": "AI智慧全面屏 6GB +64GB 亮黑色 全网通版 移动联通电信4G手机 双卡双待 双卡双待",
 "price": "3999",
 "promotionPrice": "2999",
 "originalPrice": "5999",
 "pic": "http://macro-oss.oss-cn-shenzhen.aliyuncs.com/mall/images/20180615/xiaomi.jpg",
 "sale": 999,
 "hasStock": true,
 "salecount":999,
 "putawayDate":"2021-04-01",
 "brandId": 6,
 "brandName": "小米",
 "brandImg": "http://macro-oss.oss-cn-shenzhen.aliyuncs.com/mall/images/20190129/1e34aef2a409119018a4c6258e39ecfb_222_222.png",
 "categoryId": 19,
 "categoryName": "手机通讯",
 "attrs": [
 {
 "attrId": 1,
 "attrName": "cpu",
 "attrValue": "2核"
 },
 {
 "attrId": 2,
 "attrName": "颜色",
 "attrValue": "黑色"
 }
 ]
}

PUT /product_db/_doc/2
{
 "id": "27",
 "name": "小米 10 手机",
 "keywords": "小米手机",
 "subTitle": "AI智慧全面屏 4GB +64GB 亮白色 全网通版 移动联通电信4G手机 双卡双待 双卡双待",
 "price": "2999",
 "promotionPrice": "1999",
 "originalPrice": "3999",
 "pic": "http://macro-oss.oss-cn-shenzhen.aliyuncs.com/mall/images/20180615/xiaomi.jpg",
 "sale": 999,
 "hasStock": false,
 "salecount":99,
 "putawayDate":"2021-04-02",
 "brandId": 6,
 "brandName": "小米",
 "brandImg": "http://macro-oss.oss-cn-shenzhen.aliyuncs.com/mall/images/20190129/1e34aef2a409119018a4c6258e39ecfb_222_222.png",
 "categoryId": 19,
 "categoryName": "手机通讯",
 "attrs": [
 {
 "attrId": 1,
 "attrName": "cpu",
 "attrValue": "4核"
 },
 {
 "attrId": 2,
 "attrName": "颜色",
 "attrValue": "白色"
 }
 ]
}
PUT /product_db/_doc/3
{
 "id": "28",
 "name": "小米 手机",
 "keywords": "小米手机",
 "subTitle": "AI智慧全面屏 4GB +64GB 亮蓝色 全网通版 移动联通电信4G手机 双卡双待 双卡双待",
 "price": "2999",
 "promotionPrice": "1999",
 "originalPrice": "3999",
 "pic": "http://macro-oss.oss-cn-shenzhen.aliyuncs.com/mall/images/20180615/xiaomi.jpg",
 "sale": 999,
 "hasStock": true,
 "salecount":199,
 "putawayDate":"2021-04-03",
 "brandId": 6,
 "brandName": "小米",
 "brandImg": "http://macro-oss.oss-cn-shenzhen.aliyuncs.com/mall/images/20190129/1e34aef2a409119018a4c6258e39ecfb_222_222.png",
 "categoryId": 19,
 "categoryName": "手机通讯",
 "attrs": [
 {
 "attrId": 1,
 "attrName": "cpu",
 "attrValue": "2核"
 },
 {
 "attrId": 2,
 "attrName": "颜色",
 "attrValue": "蓝色"
 }
 ]
}
PUT /product_db/_doc/4
{
 "id": "29",
 "name": "Apple iPhone 8 Plus 64GB 金色特别版 移动联通电信4G手机",
 "keywords": "苹果手机",
 "subTitle": "苹果手机 Apple产品年中狂欢节，好物尽享，美在智慧！速来 >> 勾选[保障服务][原厂保2年]，获得AppleCare+全方位服务计划，原厂延保售后无忧。",
 "price": "5999",
 "promotionPrice": "4999",
 "originalPrice": "7999",
 "pic": "http://macro-oss.oss-cn-shenzhen.aliyuncs.com/mall/images/20180615/5acc5248N6a5f81cd.jpg",
 "sale": 999,
 "hasStock": true,
 "salecount":1199,
 "putawayDate":"2021-04-04",
 "brandId": 51,
 "brandName": "苹果",
 "brandImg": "http://macro-oss.oss-cn-shenzhen.aliyuncs.com/mall/images/20180607/timg.jpg",
 "categoryId": 19,
 "categoryName": "手机通讯",
 "attrs": [
 {
 "attrId": 1,
 "attrName": "cpu",
 "attrValue": "4核"
 },
 {
 "attrId": 2,
 "attrName": "颜色",
 "attrValue": "金色"
 }
 ]
}
PUT /product_db/_doc/5
{
 "id": "30",
 "name": "HLA海澜之家简约动物印花短袖T恤",
 "keywords": "海澜之家衣服",
 "subTitle": "HLA海澜之家短袖T恤",
 "price": "199",
 "promotionPrice": "99",
 "originalPrice": "299",
 "pic": "http://macro-oss.oss-cn-shenzhen.aliyuncs.com/mall/images/20180615/5ad83a4fN6ff67ecd.jpg!cc_350x449.jpg",
 "sale": 999,
 "hasStock": true,
 "salecount":19,
 "putawayDate":"2021-04-05",
 "brandId": 50,
 "brandName": "海澜之家",
 "brandImg": "http://macro-oss.oss-cn-shenzhen.aliyuncs.com/mall/images/20190129/99d3279f1029d32b929343b09d3c72de_222_222.jpg",
 "categoryId": 8,
 "categoryName": "T恤",
 "attrs": [
 {
 "attrId": 3,
 "attrName": "尺寸",
 "attrValue": "M"
 },
 {
 "attrId": 4,
 "attrName": "颜色",
 "attrValue": "黑色"
 }
 ]
}
PUT /product_db/_doc/6
{
 "id": "31",
 "name": "HLA海澜之家蓝灰花纹圆领针织布短袖T恤",
 "keywords": "海澜之家衣服",
 "subTitle": "HLA海澜之家短袖T恤",
 "price": "299",
 "promotionPrice": "199",
 "originalPrice": "299",
 "pic": "http://macro-oss.oss-cn-shenzhen.aliyuncs.com/mall/images/20180615/5ac98b64N70acd82f.jpg!cc_350x449.jpg",
 "sale": 999,
 "hasStock": true,
 "salecount":399,
 "putawayDate":"2021-04-06",
 "brandId": 50,
 "brandName": "海澜之家",
 "brandImg": "http://macro-oss.oss-cn-shenzhen.aliyuncs.com/mall/images/20190129/99d3279f1029d32b929343b09d3c72de_222_222.jpg",
 "categoryId": 8,
 "categoryName": "T恤",
 "attrs": [
 {
 "attrId": 3,
 "attrName": "尺寸",
 "attrValue": "X"
 },
 {
 "attrId": 4,
 "attrName": "颜色",
 "attrValue": "蓝灰"
 }
 ]
}
PUT /product_db/_doc/7
{
 "id": "32",
 "name": "HLA海澜之家短袖T恤男基础款",
 "keywords": "海澜之家衣服",
 "subTitle": "HLA海澜之家短袖T恤",
 "price": "269",
 "promotionPrice": "169",
 "originalPrice": "399",
 "pic": "http://macro-oss.oss-cn-shenzhen.aliyuncs.com/mall/images/20180615/5a51eb88Na4797877.jpg",
 "sale": 999,
 "hasStock": true,
 "salecount":399,
 "putawayDate":"2021-04-07",
 "brandId": 50,
 "brandName": "海澜之家",
 "brandImg": "http://macro-oss.oss-cn-shenzhen.aliyuncs.com/mall/images/20190129/99d3279f1029d32b929343b09d3c72de_222_222.jpg",
 "categoryId": 8,
 "categoryName": "T恤",
 "attrs": [
 {
 "attrId": 3,
 "attrName": "尺寸",
 "attrValue": "L"
 },
 {
 "attrId": 4,
 "attrName": "颜色",
 "attrValue": "蓝色"
 }
 ]
}
PUT /product_db/_doc/8
{
 "id": "33",
 "name": "小米（MI）小米电视4A ",
 "keywords": "小米电视机家用电器",
 "subTitle": "小米（MI）小米电视4A 55英寸 L55M5-AZ/L55M5-AD 2GB+8GB HDR 4K超高清 人工智能网络液晶平板电视",
 "price": "2269",
 "promotionPrice": "2169",
 "originalPrice": "2399",
 "pic": "http://macro-oss.oss-cn-shenzhen.aliyuncs.com/mall/images/20180615/5b02804dN66004d73.jpg",
 "sale": 999,
 "hasStock": true,
 "salecount":132,
 "putawayDate":"2021-04-09",
 "brandId": 6,
 "brandName": "小米",
 "brandImg": "http://macro-oss.oss-cn-shenzhen.aliyuncs.com/mall/images/20190129/1e34aef2a409119018a4c6258e39ecfb_222_222.png",
 "categoryId": 35,
 "categoryName": "手机数码",
 "attrs": [
 {
 "attrId": 5,
 "attrName": "屏幕尺寸",
 "attrValue": "52"
 },
 {
 "attrId": 6,
 "attrName": "机身颜色",
 "attrValue": "黑色"
 }
 ]
}
PUT /product_db/_doc/9
{
 "id": "34",
 "name": "小米（MI）小米电视4A 65英寸",
 "keywords": "小米电视机家用电器",
 "subTitle": "小米（MI）小米电视4A 65英寸 L55M5-AZ/L55M5-AD 2GB+8GB HDR 4K超高清 人工智能网络液晶平板电视",
 "price": "3269",
 "promotionPrice": "3169",
 "originalPrice": "3399",
 "pic": "http://macro-oss.oss-cn-shenzhen.aliyuncs.com/mall/images/20180615/5b028530N51eee7d4.jpg",
 "sale": 999,
 "hasStock": true,
 "salecount":999,
 "putawayDate":"2021-04-10",
 "brandId": 6,
 "brandName": "小米",
 "brandImg": "http://macro-oss.oss-cn-shenzhen.aliyuncs.com/mall/images/20190129/1e34aef2a409119018a4c6258e39ecfb_222_222.png",
 "categoryId": 35,
 "categoryName": "手机数码",
 "attrs": [
 {
 "attrId": 5,
 "attrName": "屏幕尺寸",
 "attrValue": "65"
 },
 {
 "attrId": 6,
 "attrName": "机身颜色",
 "attrValue": "金色"
 }
 ]
}
PUT /product_db/_doc/10
{
 "id": "35",
 "name": "耐克NIKE 男子 休闲鞋 ROSHE RUN 运动鞋 511881-010黑色41码",
 "keywords": "耐克运动鞋 鞋子",
 "subTitle": "耐克NIKE 男子 休闲鞋 ROSHE RUN 运动鞋 511881-010黑色41码",
 "price": "569",
 "promotionPrice": "369",
 "originalPrice": "899",
 "pic": "http://macro-oss.oss-cn-shenzhen.aliyuncs.com/mall/images/20180615/5b235bb9Nf606460b.jpg",
 "sale": 999,
 "hasStock": true,
 "salecount":399,
 "putawayDate":"2021-04-11",
 "brandId": 58,
 "brandName": "NIKE",
 "brandImg": "http://macro-oss.oss-cn-shenzhen.aliyuncs.com/mall/images/20180615/timg (51).jpg",
 "categoryId": 29,
 "categoryName": "男鞋",
 "attrs": [
 {
 "attrId": 7,
 "attrName": "尺码",
 "attrValue": "42"
 },
 {
 "attrId": 8,
 "attrName": "颜色",
 "attrValue": "黑色"
 }
 ]
}
PUT /product_db/_doc/11
{
 "id": "36",
 "name": "耐克NIKE 男子 气垫 休闲鞋 AIR MAX 90 ESSENTIAL 运动鞋 AJ1285-101白色41码",
 "keywords": "耐克运动鞋 鞋子",
 "subTitle": "AIR MAX 90 ESSENTIAL 运动鞋 AJ1285-101白色",
 "price": "769",
 "promotionPrice": "469",
 "originalPrice": "999",
 "pic": "http://macro-oss.oss-cn-shenzhen.aliyuncs.com/mall/images/20180615/5b19403eN9f0b3cb8.jpg",
 "sale": 999,
 "hasStock": true,
 "salecount":499,
 "putawayDate":"2021-04-13",
 "brandId": 58,
 "brandName": "NIKE",
 "brandImg": "http://macro-oss.oss-cn-shenzhen.aliyuncs.com/mall/images/20180615/timg (51).jpg",
 "categoryId": 29,
 "categoryName": "男鞋",
 "attrs": [
 {
 "attrId": 7,
 "attrName": "尺码",
 "attrValue": "44"
 },
 {
 "attrId": 8,
 "attrName": "颜色",
 "attrValue": "白色"
 }
 ]
}
PUT /product_db/_doc/12
{
 "id": "37",
 "name": "(华为)HUAWEI MateBook X Pro 2019款 13.9英寸3K触控全面屏 轻薄笔记本",
 "keywords": "轻薄笔记本华为 笔记本电脑",
 "subTitle": "轻薄华为笔记本 电脑",
 "price": "4769",
 "promotionPrice": "4469",
 "originalPrice": "4999",
 "pic": "http://tuling-mall.oss-cn-shenzhen.aliyuncs.com/tulingmall/images/20200317/800_800_1555752016264mp.png",
 "sale": 999,
 "hasStock": true,
 "salecount":699,
 "putawayDate":"2021-04-14",
 "brandId": 3,
 "brandName": "华为",
 "brandImg": "http://macro-oss.oss-cn-shenzhen.aliyuncs.com/mall/images/20190129/17f2dd9756d9d333bee8e60ce8c03e4c_222_222.jpg",
 "categoryId": 19,
 "categoryName": "手机通讯",
 "attrs": [
 {
 "attrId": 9,
 "attrName": "容量",
 "attrValue": "16G"
 },
 {
 "attrId": 10,
 "attrName": "网络",
 "attrValue": "4G"
 }
 ]
}
PUT /product_db/_doc/13
{
 "id": "38",
 "name": "华为nova6se 手机 绮境森林 全网通（8G+128G)",
 "keywords": "轻薄笔记本华为 手机",
 "subTitle": "华为nova6se 手机",
 "price": "6769",
 "promotionPrice": "6469",
 "originalPrice": "6999",
 "pic": "http://macro-oss.oss-cn-shenzhen.aliyuncs.com/mall/images/20180607/5ac1bf58Ndefaac16.jpg",
 "sale": 999,
 "hasStock": true,
 "salecount":899,
 "putawayDate":"2021-04-15",
 "brandId": 3,
 "brandName": "华为",
 "brandImg": "http://macro-oss.oss-cn-shenzhen.aliyuncs.com/mall/images/20190129/17f2dd9756d9d333bee8e60ce8c03e4c_222_222.jpg",
 "categoryId": 19,
 "categoryName": "手机通讯",
 "attrs": [
 {
 "attrId": 9,
 "attrName": "容量",
 "attrValue": "64G"
 },
 {
 "attrId": 10,
 "attrName": "网络",
 "attrValue": "5G"
 }
```

```
 ]
}
PUT /product_db/_doc/14
{
 "id": "39",
 "name": "iPhone7/6s/8钢化膜苹果8Plus全屏复盖抗蓝光防窥防偷看手机膜",
 "keywords": "手机膜",
 "subTitle": "iPhone7/6s/8钢化膜苹果8Plus全屏复盖抗蓝光防窥防偷看手机膜",
 "price": "29",
 "promotionPrice": "39",
 "originalPrice": "49",
 "pic": "http://tuling-mall.oss-cn-shenzhen.aliyuncs.com/tulingmall/images/20200311/6df99dab78bb2014.jpg",
 "sale": 999,
 "hasStock": true,
 "salecount":799,
 "putawayDate":"2021-04-16",
 "brandId": 51,
 "brandName": "苹果",
 "brandImg": "http://tuling-mall.oss-cn-shenzhen.aliyuncs.com/tulingmall/images/20200311/2b84746650fc122d67749a876c453619.png",
 "categoryId": 30,
 "categoryName": "手机配件",
 "attrs": [
 {
 "attrId": 11,
 "attrName": "手机膜-材料",
 "attrValue": "钢化"
 },
 {
 "attrId": 12,
 "attrName": "手机膜-颜色",
 "attrValue": "白色"
 }
 ]
}

PUT /product_db/_doc/15
{
 "id": "40",
 "name": "七匹狼短袖T恤男纯棉舒适春夏修身运动休闲短袖三条装 圆领3条装",
 "keywords": "七匹狼服装 衣服",
 "subTitle": "七匹狼短袖T恤男纯棉舒适春夏修身运动休闲短袖三条装 圆领3条装",
 "price": "129",
 "promotionPrice": "139",
 "originalPrice": "149",
 "pic": "http://tuling-mall.oss-cn-shenzhen.aliyuncs.com/tulingmall/images/20200311/19e846e727dff337.jpg",
 "sale": 999,
 "hasStock": true,
 "salecount":199,
 "putawayDate":"2021-04-20",
 "brandId": 49,
 "brandName": "七匹狼",
 "brandImg": "http://macro-oss.oss-cn-shenzhen.aliyuncs.com/mall/images/20190129/18d8bc3eb13533fab466d702a0d3fd1f40345bcd.jpg",
 "categoryId": 8,
 "categoryName": "T恤",
 "attrs": [
 {
 "attrId": 3,
 "attrName": "尺寸",
 "attrValue": "M"
 },
 {
 "attrId": 4,
 "attrName": "颜色",
 "attrValue": "白色"
 }
 ]
}
PUT /product_db/_doc/16
{
 "id": "41",
 "name": "华为P40 Pro手机",
 "keywords": "华为手机",
 "subTitle": "华为P40 Pro手机",
 "price": "2129",
 "promotionPrice": "2139",
 "originalPrice": "2149",
 "pic": "http://macro-oss.oss-cn-shenzhen.aliyuncs.com/mall/images/20180607/5ac1bf58Ndefaac16.jpg",
 "sale": 999,
 "hasStock": true,
 "salecount":199,
 "putawayDate":"2021-05-03",
 "brandId": 3,
 "brandName": "华为",
 "brandImg": "http://macro-oss.oss-cn-shenzhen.aliyuncs.com/mall/images/20190129/17f2dd9756d9d333bee8e60ce8c03e4c_222_222.jpg",
 "categoryId": 19,
 "categoryName": "手机通讯",
 "attrs": [
 {
 "attrId": 9,
 "attrName": "容量",
 "attrValue": "128G"
 },
 {
 "attrId": 10,
 "attrName": "网络",
 "attrValue": "5G"
 }
 ]
}
PUT /product_db/_doc/17
{
 "id": "42",
 "name": "朵唯智能手机 4G全网通 老人学生双卡双待手机",
 "keywords": "朵唯手机",
 "subTitle": "朵唯手机后置双摄，国产虎贲芯片！优化散热结构！浅薄机身！朵唯4月特惠！",
 "price": "3129",
 "promotionPrice": "3139",
 "originalPrice": "3249",
 "pic": "http://macro-oss.oss-cn-shenzhen.aliyuncs.com/mall/images/20180615/xiaomi.jpg",
 "sale": 999,
 "hasStock": true,
 "salecount":1199,
 "putawayDate":"2021-06-01",
 "brandId": 59,
 "brandName": "朵唯",
 "brandImg": "http://tuling-mall.oss-cn-shenzhen.aliyuncs.com/tulingmall/images/20200311/2b84746650fc122d67749a876c453619.png",
 "categoryId": 19,
 "categoryName": "手机通讯",
 "attrs": [
 {
 "attrId": 9,
 "attrName": "容量",
 "attrValue": "32G"
 },
 {
 "attrId": 10,
 "attrName": "网络",
 "attrValue": "4G"
 }
 ]
}

```

## 1.4 构建DSL语句实现商品搜索

ES高级查询语法Query DSL实战  ElasticSearch搜索技术深入与聚合查询实战

```
POST /product_db/_doc/_search
{
 "from": 0,
 "size": 8,
 "query": {
 "bool": {
 "must": [
 {
 "match": {
  "name": {
  "query": "手机"
  }
  }
  }
  ],
  "filter": [
  {
  "term": {
  "hasStock": {
  "value": true
  }
  }
  },
  {
  "range": {
  "price": {
  "from": "1",
  "to": "5000"
  }
  }
  }
  ]
  }
  },"sort": [
  {
  "salecount": {
  "order": "asc"
  }
  }
  ],
  "aggregations": {
```

```
"brand_agg": {
"terms": {
"field": "brandId",
"size": 50
},
"aggregations": {
"brand_name_agg": {
"terms": {
"field": "brandName"
}
},
"brand_img_agg": {
"terms": {
"field": "brandImg"
}
}
}
},
"category_agg": {
"terms": {
"field": "categoryId",
"size": 50,
"min_doc_count": 1
},
"aggregations": {
"category_name_agg": {
"terms": {
"field": "categoryName"
}
}
}
},
"attr_agg": {
"nested": {
"path": "attrs"
},
"aggregations": {
"attr_id_agg": {
"terms": {
"field": "attrs.attrId"
},
"aggregations": {
"attr_name_agg": {
"terms": {
"field": "attrs.attrName"
}
},
"attr_value_agg": {
"terms": {
"field": "attrs.attrValue"
}
}
}
}
}
}
},
"highlight": {
 "pre_tags": [
 "<b style='color:red'>"
```

```
  ],
  "post_tags": [
  "</b>"
  ],
  "fields": {
  "name": {}
  }
  }
 }
```

```
GET product_db/_search
{
 "from": 0,
 "size": 20,
 "query": {
 "bool": {
 "must": [
 {
 "multi_match": {
  "query": "手机",
  "fields": [
  "name",
  "keywords",
  "subTitle"
  ]
  }
  }
  ],
  "filter": [
  {
  "term": {
  "hasStock": "true"
  }
  },
  {
  "range": {
  "price": {
  "gte": 2000,
  "lte": 5000
  }
  }
  }
  ]
  }
  },
  "aggs": {
  "brandId_aggs": {
  "terms": {
  "field": "brandId",
  "size": 10
  },
  "aggs": {
  "brandName_aggs": {
  "terms": {
  "field": "brandName"
  }
  },
"brandImg_aggs": {
"terms": {
"field": "brandImg"
}
}
}
},
"categoryId_aggs": {
"terms": {
"field": "categoryId",
"size": 10
},
"aggs": {
"categoryName_aggs": {
"terms": {
"field": "categoryName"
}
}
}
},
"attrs_aggs": {
"nested": {
"path": "attrs"
},
"aggs": {
"attrId_aggs": {
"terms": {
"field": "attrs.attrId"
},
"aggs": {
"attrName_aggs": {
"terms": {
"field": "attrs.attrName"
}
},
"attrValue_aggs": {
"terms": {
"field": "attrs.attrValue"
}
}
}
}
}
}
},
"sort": [
{
"salecount": {
"order": "desc"
}
}
],
 "highlight": {
 "fields": {
 "*": {}
 }
 }
}
```

## 1.5 Java代码实现商品搜索功能

1）引入依赖

```xml
<dependency>
 <groupId>org.springframework.boot</groupId>
 <artifactId>spring-boot-starter-data-elasticsearch</artifactId>
</dependency>
```

2）核心代码

```java
@Qualifier("restHighLevelClient")
@Autowired
private RestHighLevelClient client;


@Override
public ESResponseResult search(ESRequestParam param) {

 try {
 //1、构建检索对象-封装请求相关参数信息
 SearchRequest searchRequest = startBuildRequestParam(param);

 //2、进行检索操作
 SearchResponse response = client.search(searchRequest, RequestOptions.DEFAULT);

 //3、分析响应数据，封装成指定的格式
 ESResponseResult responseResult = startBuildResponseResult(response, param);
 return responseResult;
 } catch (Exception e) {
 e.printStackTrace();
 }

 return null;

}

/**
 * 封装请求参数信息
 * 关键字查询、根据属性、分类、品牌、价格区间、是否有库存等进行过滤、分页、高亮、以及聚合统计品牌分类属性
 *
 */
private SearchRequest startBuildRequestParam(ESRequestParam param) {

 SearchSourceBuilder searchSourceBuilder = new SearchSourceBuilder();

 /**
 * 关键字查询、根据属性、分类、品牌、价格区间、是否有库存等进行过滤、分页、高亮、以及聚合统计品牌分类属性
 */
 BoolQueryBuilder boolQueryBuilder = new BoolQueryBuilder();

 //1、
 if (!StringUtils.isEmpty(param.getKeyword())) {
 //单字段查询
 //boolQueryBuilder.must(QueryBuilders.matchQuery("name", param.getKeyword()));
 //多字段查询
 boolQueryBuilder.must(QueryBuilders.multiMatchQuery(param.getKeyword(),"name","keywords","subTitle"));
 }

 //2、根据类目ID进行过滤
 if (null != param.getCategoryId()) {
 boolQueryBuilder.filter(QueryBuilders.termQuery("categoryId", param.getCategoryId()));
 }

 //3、根据品牌ID进行过滤
 if (null != param.getBrandId() && param.getBrandId().size() > 0) {
 boolQueryBuilder.filter(QueryBuilders.termsQuery("brandId", param.getBrandId()));
 }

 //4、根据属性进行相关过滤
 if (param.getAttrs() != null && param.getAttrs().size() > 0) {

 param.getAttrs().forEach(item -> {
 //attrs=1_白色&2_4核
 BoolQueryBuilder boolQuery = QueryBuilders.boolQuery();

 //attrs=1_64G
 String[] s = item.split("_");
 String attrId = s[0];
 String[] attrValues = s[1].split(":");//这个属性检索用的值
 boolQuery.must(QueryBuilders.termQuery("attrs.attrId", attrId));
 boolQuery.must(QueryBuilders.termsQuery("attrs.attrValue", attrValues));

 NestedQueryBuilder nestedQueryBuilder = QueryBuilders.nestedQuery("attrs", boolQuery, ScoreMode.None);
 boolQueryBuilder.filter(nestedQueryBuilder);
 });

 }

 //5、是否有库存
 if (null != param.getHasStock()) {
 boolQueryBuilder.filter(QueryBuilders.termQuery("hasStock", param.getHasStock() == 1));
 }

 /* if (price.length == 2) {
 //price: _5000
 if (param.getPrice().startsWith("_")) {
 rangeQueryBuilder.lte(price[1]);
 }
 else{
 //price: 1_5000
 rangeQueryBuilder.gte(price[0]).lte(price[1]);
 }

} else if (price.length == 1) {
 //price: 1_
 if (param.getPrice().endsWith("_")) {
 rangeQueryBuilder.gte(price[0]);
 }
}*/

 //6、根据价格过滤
 if (!StringUtils.isEmpty(param.getPrice())) {
 //价格的输入形式为：10-100（起始价格和最终价格）或-100（不指定起始价格）或10-（不限制最终价格）
 RangeQueryBuilder rangeQueryBuilder = QueryBuilders.rangeQuery("price");
 String[] price = param.getPrice().split("_");
 if (price.length == 2) {
 rangeQueryBuilder.gte(price[0]).lte(price[1]);
 } else if (price.length == 1) {
 if (param.getPrice().startsWith("_")) {
 rangeQueryBuilder.lte(price[1]);
 }
 if (param.getPrice().endsWith("_")) {
 rangeQueryBuilder.gte(price[0]);
}
}
boolQueryBuilder.filter(rangeQueryBuilder);
}

//封装所有查询条件
searchSourceBuilder.query(boolQueryBuilder);


/**
* 实现排序、高亮、分页操作
*/

//排序
//页面传入的参数值形式 sort=price_asc/desc
if (!StringUtils.isEmpty(param.getSort())) {
String sort = param.getSort();
String[] sortFileds = sort.split("_");
System.out.println("sortFileds:"+sortFileds.length);
if(!StringUtils.isEmpty(sortFileds[0])){
SortOrder sortOrder = "asc".equalsIgnoreCase(sortFileds[1]) ? SortOrder.ASC : SortOrder.DESC;
searchSourceBuilder.sort(sortFileds[0], sortOrder);
}
}

//分页查询
searchSourceBuilder.from((param.getPageNum() - 1) * SearchConstant.PAGE_SIZE);
searchSourceBuilder.size(SearchConstant.PAGE_SIZE);

//高亮显示
if (!StringUtils.isEmpty(param.getKeyword())) {

HighlightBuilder highlightBuilder = new HighlightBuilder();
highlightBuilder.field("name");
highlightBuilder.preTags("<b style='color:red'>");
highlightBuilder.postTags("</b>");

searchSourceBuilder.highlighter(highlightBuilder);
}


/**
* 对品牌、分类信息、属性信息进行聚合分析
*/
//1. 按照品牌进行聚合
TermsAggregationBuilder brand_agg = AggregationBuilders.terms("brand_agg");
brand_agg.field("brandId").size(50);


//1.1 品牌的子聚合-品牌名聚合
brand_agg.subAggregation(AggregationBuilders.terms("brand_name_agg")
.field("brandName").size(1));
//1.2 品牌的子聚合-品牌图片聚合
brand_agg.subAggregation(AggregationBuilders.terms("brand_img_agg")
.field("brandImg").size(1));

searchSourceBuilder.aggregation(brand_agg);

//2. 按照分类信息进行聚合
TermsAggregationBuilder category_agg = AggregationBuilders.terms("category_agg");
category_agg.field("categoryId").size(50);
```

```

 category_agg.subAggregation(AggregationBuilders.terms("category_name_agg").field("categoryName").size(1));

 searchSourceBuilder.aggregation(category_agg);

 //2. 按照属性信息进行聚合
 NestedAggregationBuilder attr_agg = AggregationBuilders.nested("attr_agg", "attrs");
 //2.1 按照属性ID进行聚合
 TermsAggregationBuilder attr_id_agg = AggregationBuilders.terms("attr_id_agg").field("attrs.attrId");
 attr_agg.subAggregation(attr_id_agg);
 //2.1.1 在每个属性ID下，按照属性名进行聚合
 attr_id_agg.subAggregation(AggregationBuilders.terms("attr_name_agg").field("attrs.attrName").size(1));
 //2.1.1 在每个属性ID下，按照属性值进行聚合
 attr_id_agg.subAggregation(AggregationBuilders.terms("attr_value_agg").field("attrs.attrValue").size(50));
 searchSourceBuilder.aggregation(attr_agg);

 System.out.println("构建的DSL语句 {}:"+ searchSourceBuilder.toString());

 SearchRequest searchRequest = new SearchRequest(new String[]{SearchConstant.INDEX_NAME}, searchSourceBuilder);

 return searchRequest;
}


/**
 * 封装查询到的结果信息
 * 关键字查询、根据属性、分类、品牌、价格区间、是否有库存等进行过滤、分页、高亮、以及聚合统计品牌分类属性
 */
private ESResponseResult startBuildResponseResult(SearchResponse response, ESRequestParam param) {

 ESResponseResult result = new ESResponseResult();

 //1、获取查询到的商品信息
 SearchHits hits = response.getHits();

 List<EsProduct> esModels = new ArrayList<>();
 //2、遍历所有商品信息
 if (hits.getHits() != null && hits.getHits().length > 0) {
 for (SearchHit hit : hits.getHits()) {
 String sourceAsString = hit.getSourceAsString();
 EsProduct esModel = JSON.parseObject(sourceAsString, EsProduct.class);

 //2.1 判断是否按关键字检索，若是就显示高亮，否则不显示
 if (!StringUtils.isEmpty(param.getKeyword())) {
 //2.2 拿到高亮信息显示标题
 HighlightField name = hit.getHighlightFields().get("name");
 //2.3 判断name中是否含有查询的关键字(因为是多字段查询，因此可能不包含指定的关键字，假设不包含则显示原始name字段的信息)
 String nameValue = name!=null ? name.getFragments()[0].string() : esModel.getName();
 esModel.setName(nameValue);
 }
 esModels.add(esModel);
 }
 }
 result.setProducts(esModels);

 //3、当前商品涉及到的所有品牌信息，小米手机和小米电脑都属于小米品牌，过滤重复品牌信息
 Set<ESResponseResult.BrandVo> brandVos = new LinkedHashSet<>();
 //获取到品牌的聚合
 ParsedLongTerms brandAgg = response.getAggregations().get("brand_agg");
 for (Terms.Bucket bucket : brandAgg.getBuckets()) {
 ESResponseResult.BrandVo brandVo = new ESResponseResult.BrandVo();
```

```

//获取品牌的id
long brandId = bucket.getKeyAsNumber().longValue();
brandVo.setBrandId(brandId);

//获取品牌的名字
ParsedStringTerms brandNameAgg = bucket.getAggregations().get("brand_name_agg");
String brandName = brandNameAgg.getBuckets().get(0).getKeyAsString();
brandVo.setBrandName(brandName);

//获取品牌的LOGO
ParsedStringTerms brandImgAgg = bucket.getAggregations().get("brand_img_agg");
String brandImg = brandImgAgg.getBuckets().get(0).getKeyAsString();
brandVo.setBrandImg(brandImg);
System.out.println("brandId:"+brandId+"brandName:"+brandName+"brandImg");
brandVos.add(brandVo);
}
System.out.println("brandVos.size:"+brandVos.size());
result.setBrands(brandVos);


//4、当前商品相关的所有类目信息
//获取到分类的聚合
List<ESResponseResult.categoryVo> categoryVos = new ArrayList<>();

ParsedLongTerms categoryAgg = response.getAggregations().get("category_agg");


for (Terms.Bucket bucket : categoryAgg.getBuckets()) {
ESResponseResult.categoryVo categoryVo = new ESResponseResult.categoryVo();
//获取分类id
String keyAsString = bucket.getKeyAsString();
categoryVo.setCategoryId(Long.parseLong(keyAsString));

//获取分类名
ParsedStringTerms categoryNameAgg = bucket.getAggregations().get("category_name_agg");
String categoryName = categoryNameAgg.getBuckets().get(0).getKeyAsString();
categoryVo.setCategoryName(categoryName);
categoryVos.add(categoryVo);
}

result.setCategorys(categoryVos);



//5、获取商品相关的所有属性信息
List<ESResponseResult.AttrVo> attrVos = new ArrayList<>();
//获取属性信息的聚合
ParsedNested attrsAgg = response.getAggregations().get("attr_agg");
ParsedLongTerms attrIdAgg = attrsAgg.getAggregations().get("attr_id_agg");
for (Terms.Bucket bucket : attrIdAgg.getBuckets()) {
ESResponseResult.AttrVo attrVo = new ESResponseResult.AttrVo();
//获取属性ID值
long attrId = bucket.getKeyAsNumber().longValue();
attrVo.setAttrId(attrId);

//获取属性的名字
ParsedStringTerms attrNameAgg = bucket.getAggregations().get("attr_name_agg");
String attrName = attrNameAgg.getBuckets().get(0).getKeyAsString();
attrVo.setAttrName(attrName);

```

```
//获取属性的值
ParsedStringTerms attrValueAgg = bucket.getAggregations().get("attr_value_agg");
List<String> attrValues = attrValueAgg.getBuckets().stream().map(item -> item.getKeyAsString()).collect(Collectors.toList());
attrVo.setAttrValue(attrValues);

attrVos.add(attrVo);
}

result.setAttrs(attrVos);

//6、进行分页操作
result.setPageNum(param.getPageNum());
//获取总记录数
long total = hits.getTotalHits().value;
result.setTotal(total);

//计算总页码
int totalPages = (int) total % SearchConstant.PAGE_SIZE == 0 ?
(int) total / SearchConstant.PAGE_SIZE : ((int) total / SearchConstant.PAGE_SIZE + 1);
result.setTotalPages(totalPages);

List<Integer> pageNavs = new ArrayList<>();
for (int i = 1; i <= totalPages; i++) {
pageNavs.add(i);
}
result.setPageNavs(pageNavs);

return result;
}

```

测试

http://localhost:8054/searchList?price=1_5000&keyword=%E6%89%8B%E6%9C%BA&sort=salecount_asc&hasStock=1&pageNum=1&pageSize=20&categoryIc

# 2. ES跨集群数据迁移

## 2.1 elasticsearch-dump

**适用场景**

适合数据量不大，迁移索引个数不多的场景

**使用方式**

elasticsearch-dump是一款开源的ES数据迁移工具，github地址

1）安装elasticsearch-dump

elasticsearch-dump使用node.js开发，可使用npm包管理工具直接安装：

```
npm install elasticdump -g
```

2) 主要参数说明:

- --input: 源地址，可为ES集群URL、文件或stdin,可指定索引，格式为：{protocol}://{host}:{port}/{index}
- --input-index: 源ES集群中的索引
- --output: 目标地址，可为ES集群地址URL、文件或stdout，可指定索引，格式为：{protocol}://{host}:{port}/{index}
- --output-index: 目标ES集群的索引
- --type: 迁移类型，默认为data,表明只迁移数据，可选settings, analyzer, data, mapping, alias
- --limit：每次向目标ES集群写入数据的条数，不可设置的过大，以免bulk队列写满

3) 迁移单个索引

以下操作通过elasticdump命令将集群172.16.0.39中的company database索引迁移至集群172.16.0.20。注意第一条命令先将索引的settings先迁移，如果直接迁移mapping或者data将失去原有集群中索引的配置信息如分片数量和副本数量等，当然也可以直接在目标集群中将索引创建完毕后再同步mapping与data

```
elasticdump --input=http://172.16.0.39:9200/companydatabase --output=http://172.16.0.20:9200/companydatabase --type=settings
elasticdump --input=http://172.16.0.39:9200/companydatabase --output=http://172.16.0.20:9200/companydatabase --type=mapping
elasticdump --input=http://172.16.0.39:9200/companydatabase --output=http://172.16.0.20:9200/companydatabase --type=data
```

4) 迁移所有索引

以下操作通过elasticdump命令将将集群172.16.0.39中的所有索引迁移至集群172.16.0.20。 注意此操作并不能迁移索引的配置如分片数量和副本数量，必须对每个索引单独进行配置的迁移，或者直接在目标集群中将索引创建完毕后再迁移数据

```
elasticdump --input=http://172.16.0.39:9200 --output=http://172.16.0.20:9200
```

## 2.2 snapshot

**适用场景**

适用数据量大的场景

**使用方式**

snapshot api是Elasticsearch用于对数据进行备份和恢复的一组api接口，可以通过snapshot api进行跨集群的数据迁移，原理就是从源ES集群创建数据快照，然后在目标ES集群中进行恢复。需要注意ES的版本问题：

- 目标ES集群的主版本号(如5.6.4中的5为主版本号)要大于等于源ES集群的主版本号;
- 1.x版本的集群创建的快照不能在5.x版本中恢复;

1) 源ES集群中创建repository

创建快照前必须先创建repository仓库，一个repository仓库可以包含多份快照文件，repository主要有一下几种类型

- fs: 共享文件系统，将快照文件存放于文件系统中
- url: 指定文件系统的URL路径，支持协议：http,https,ftp,file,jar
- s3: AWS S3对象存储,快照存放于S3中，以插件形式支持
- hdfs: 快照存放于hdfs中，以插件形式支持
- cos: 快照存放于腾讯云COS对象存储中，以插件形式支持

如果需要从自建ES集群迁移至腾讯云的ES集群，可以直接使用fs类型仓库，注意需要在Elasticsearch配置文件elasticsearch.yml设置仓库路径：

```
path.repo: ["/usr/local/services/test"]
```

之后调用snapshot api创建repository：

```
curl -XPUT http://172.16.0.39:9200/_snapshot/my_backup -H 'Content-Type: application/json' -d '{
 "type": "fs",
 "settings": {
 "location": "/usr/local/services/test"
 "compress": true
 }
}'
```

如果需要从其它云厂商的ES集群迁移至腾讯云ES集群，或者腾讯云内部的ES集群迁移，可以使用对应云厂商他提供的仓库类型，如AWS的S3, 阿里云的OSS，腾讯云的COS等

```
curl -XPUT http://172.16.0.39:9200/_snapshot/my_s3_repository
{
 "type": "s3",
 "settings": {
 "bucket": "my_bucket_name",
 "region": "us-west"
 }
}
```

2) 源ES集群中创建snapshot

调用snapshot api在创建好的仓库中创建快照

```
curl -XPUT http://172.16.0.39:9200/_snapshot/my_backup/snapshot_1?wait_for_completion=true
```

创建快照可以指定索引，也可以指定快照中包含哪些内容，具体的api接口参数可以查阅官方文档

3 . 目标ES集群中创建repository

目标ES集群中创建仓库和在源ES集群中创建仓库类似，用户可在腾讯云上创建COS对象bucket， 将仓库将在COS的某个bucket下。

4 . 移动源ES集群snapshot至目标ES集群的仓库

把源ES集群创建好的snapshot上传至目标ES集群创建好的仓库中

5 . 从快照恢复

```
curl -XPUT http://172.16.0.20:9200/_snapshot/my_backup/snapshot_1/_restore
```

6 . 查看快照恢复状态

```
curl http://172.16.0.20:9200/_snapshot/_status
```

## 2.3 reindex

reindex是Elasticsearch提供的一个api接口，可以把数据从源ES集群导入到当前的ES集群，同样实现了数据的迁移。reindex可以做跨索引、跨集群的数据迁移，应用于大量集群数据的迁移。

1) 配置reindex.remote.whitelist参数

需要在目标ES集群中配置该参数，指明能够reindex的远程集群的白名单

2) 调用reindex api

以下操作表示从源ES集群中查询名为test1的索引，查询条件为title字段为elasticsearch，将结果写入当前集群的test2索引

```
POST _reindex
{
 "source": {
 "remote": {
 "host": "http://172.16.0.39:9200"
 },
 "index": "test1",
 "query": {
 "match": {
 "title": "elasticsearch"
 }
 }
 },
 "dest": {
 "index": "test2"
 }
}
```

## 2.4 logstash

logstash支持从一个ES集群中读取数据然后写入到另一个ES集群，因此可以使用logstash进行数据迁移，具体的配置文件如下：

```
input {
elasticsearch {
hosts => ["http://172.16.0.39:9200"]
index => "*"
docinfo => true
}
}
output {
elasticsearch {
hosts => ["http://172.16.0.20:9200"]
index => "%{[@metadata][_index]}"
}
}
```

上述配置文件将源ES集群的所有索引同步到目标集群中，当然可以设置只同步指定的索引，logstash的更多功能可查阅logstash官方文档

**example:**

```
input {
    elasticsearch {
    #action => "index" #动作 固定值不需要修改
    hosts => "10.98.8.151:9200" #ElasticSearch主机
    index => "*" #索引名称
    user => "elastic" #用户名
    password =>"xxxxx" #密码
    size => 5000
```

```
   scroll => "50m"
    docinfo => true
    #会提取ES文档的元信息metadata
    }
}
filter {
    #Only matched data are send to output.
}
output {
    #stdout {
    #      codec => rubydebug
    #}
    elasticsearch {
        #action => "index" #动作 固定值不需要修改
        hosts => "10.88.8.5:9200" #ElasticSearch主机
        index => "%{[@metadata][_index]}" #索引名称
        document_type => "%{[@metadata][_type]}"
        document_id => "%{[@metadata][_id]}"
        user => "elastic" #用户名
        password => "elastic" #密码
    }
}
```

**总结**

1. elasticsearch-dump和logstash做跨集群数据迁移时，都要求用于执行迁移任务的机器可以同时访问到两个集群，不然网络无法连通的情况下就无法实现迁移。而使用snapshot的方式没有这个限制，因为snapshot方式是完全离线的。因此elasticsearch-dump和logstash迁移方式更适合于源ES集群和目标ES集群处于同一网络的情况下进行迁移，而需要跨云厂商的迁移，比如从阿里云ES集群迁移至腾讯云ES集群，可以选择使用snapshot的方式进行迁移，当然也可以通过打通网络实现集群互通，但是成本较高。
2. elasticsearchdump工具和mysql数据库用于做数据备份的工具mysqldump工具类似，都是逻辑备份，需要将数据一条一条导出后再执行导入，所以适合数据量小的场景下进行迁移；
3. snapshot的方式适合数据量大的场景下进行迁移。