

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](#)

- 1 文档：[5 电商项目ElasticSearch高性能搜索实...](#)
- 2 链接：<http://note.youdao.com/noteshare?id=48251867771266d6b9d2d4d1b5b14b27&sub=95D841A7B8DC40718C6DDDB35199F055>

1.ElasticSearch回顾

1.1 业务场景——图灵商城商品搜索

1.2 文档建模

1.3 索引文档

1.4 构建DSL语句实现商品搜索

1.5 Java代码实现商品搜索功能

2. ES跨集群数据迁移

2.1 elasticsearch-dump

2.2 snapshot

2.3 reindex

2.4 logstash

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/>

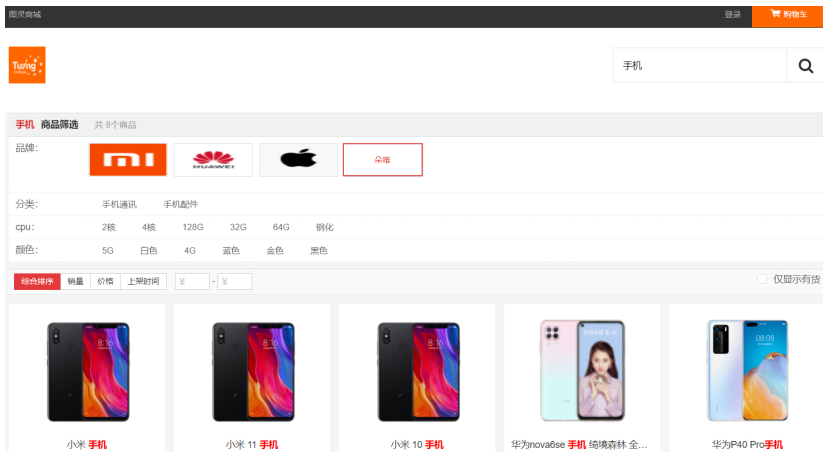
下载地址：<https://www.elastic.co/cn/downloads/past-releases#elasticsearch>

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

```

1 {
2   "id": "26",
3   "name": "小米 11 手机",
4   "keywords": "小米手机",
5   "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",
6   "price": "3999",
7   "promotionPrice": "2999",
8   "originalPrice": "5999",
9   "pic": "http://macro-oss.oss-cn-shenzhen.aliyuncs.com/mall/images/20180615/xiaomi.jpg",
10  "sale": 999,
11  "hasStock": true,
12  "salecount": 999,
13  "putawayDate": "2021-04-01",
14  "brandId": 6,
15  "brandName": "小米",
16  "brandImg": "http://macro-oss.oss-cn-shenzhen.aliyuncs.com/mall/images/20190129/1e34aef2a409119018a4c6258e39ecfb_22_222.png",
17  "categoryId": 19,
18  "categoryName": "手机通讯",
19  "attrs": [
20    {
21      "attrId": 1,
22      "attrName": "cpu",
23      "attrValue": "2核"
24    },
25    {
26      "attrId": 2,
27      "attrName": "颜色",
28      "attrValue": "黑色"
29    }
30  ]
31 }
32
33 {
34   "id": "30",
35   "name": "HLA海澜之家简约动物印花短袖T恤",
36   "keywords": "海澜之家衣服",
37   "subTitle": "HLA海澜之家短袖T恤",
38   "price": "199",
39   "promotionPrice": "99",
40   "originalPrice": "299",
41   "pic": "http://macro-oss.oss-cn-shenzhen.aliyuncs.com/mall/images/20180615/5ad83a4fN6ff67ecd.jpg!cc_350x449.jpg",
42   "sale": 999,

```

```

43  "hasStock": true,
44  "salecount":19,
45  "putawayDate":"2021-04-05",
46  "brandId": 50,
47  "brandName": "海澜之家",
48  "brandImg": "http://macro-oss.oss-cn-shenzhen.aliyuncs.com/mall/images/20190129/99d3279f1029d32b929343b09d3c72de_22_222.jpg",
49  "categoryId": 8,
50  "categoryName": "T恤",
51  "attrs": [
52  {
53    "attrId": 3,
54    "attrName": "尺寸",
55    "attrValue": "M"
56  },
57  {
58    "attrId": 4,
59    "attrName": "颜色",
60    "attrValue": "黑色"
61  }
62  ]
63  }

```

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

```

1  PUT product_db
2  {
3    "mappings": {
4      "properties": {
5        "id": {
6          "type": "long"
7        },
8        "name": {
9          "type": "text",
10         "analyzer": "ik_max_word"
11       },
12       "keywords": {
13         "type": "text",
14         "analyzer": "ik_max_word"
15       },
16       "subTitle": {
17         "type": "text",
18         "analyzer": "ik_max_word"
19       },
20       "salecount":{
21         "type": "long"

```

```
22 },
23 "putawayDate":{
24   "type": "date"
25 },
26 "price": {
27   "type": "double"
28 },
29
30 "promotionPrice": {
31   "type": "keyword"
32 },
33 "originalPrice": {
34   "type": "keyword"
35 },
36 "pic": {
37   "type": "keyword"
38 },
39 "sale": {
40   "type": "long"
41 },
42 "hasStock": {
43   "type": "boolean"
44 },
45 "brandId": {
46   "type": "long"
47 },
48 "brandName": {
49   "type": "keyword"
50 },
51 "brandImg": {
52   "type": "keyword"
53 },
54 "categoryId": {
55   "type": "long"
56 },
57 "categoryName": {
58   "type": "keyword"
59 },
60 "attrs": {
61   "type": "nested",
62   "properties": {
63     "attrId": {
64       "type": "long"
65     },
66     "attrName": {
67       "type": "keyword"
68     },
69     "attrValue": {
70       "type": "keyword"
71     }
72   }
73 }
74 }
75 }
76 }
77 }
```

1.3 索引文档

可以利用canal进行数据同步

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

```
2 {
3   "id": "26",
4   "name": "小米 11 手机",
5   "keywords": "小米手机",
6   "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",
7   "price": "3999",
8   "promotionPrice": "2999",
9   "originalPrice": "5999",
10  "pic": "http://macro-oss.oss-cn-shenzhen.aliyuncs.com/mall/images/20180615/xiaomi.jpg",
11  "sale": 999,
12  "hasStock": true,
13  "salecount": 999,
14  "putawayDate": "2021-04-01",
15  "brandId": 6,
16  "brandName": "小米",
17  "brandImg": "http://macro-oss.oss-cn-shenzhen.aliyuncs.com/mall/images/20190129/1e34aef2a409119018a4c6258e39ecfb_22_222.png",
18  "categoryId": 19,
19  "categoryName": "手机通讯",
20  "attrs": [
21    {
22      "attrId": 1,
23      "attrName": "cpu",
24      "attrValue": "2核"
25    },
26    {
27      "attrId": 2,
28      "attrName": "颜色",
29      "attrValue": "黑色"
30    }
31  ]
32 }
33
34 PUT /product_db/_doc/2
35 {
36   "id": "27",
37   "name": "小米 10 手机",
38   "keywords": "小米手机",
39   "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",
40   "price": "2999",
41   "promotionPrice": "1999",
42   "originalPrice": "3999",
43   "pic": "http://macro-oss.oss-cn-shenzhen.aliyuncs.com/mall/images/20180615/xiaomi.jpg",
44   "sale": 999,
45   "hasStock": false,
46   "salecount": 99,
47   "putawayDate": "2021-04-02",
48   "brandId": 6,
49   "brandName": "小米",
50   "brandImg": "http://macro-oss.oss-cn-shenzhen.aliyuncs.com/mall/images/20190129/1e34aef2a409119018a4c6258e39ecfb_22_222.png",
51   "categoryId": 19,
52   "categoryName": "手机通讯",
53   "attrs": [
54     {
55       "attrId": 1,
56       "attrName": "cpu",
57       "attrValue": "4核"
58     },
59     {
60       "attrId": 2,
```

```
61  "attrName": "颜色",
62  "attrValue": "白色"
63  }
64  ]
65  }
66  PUT /product_db/_doc/3
67  {
68  "id": "28",
69  "name": "小米 手机",
70  "keywords": "小米手机",
71  "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",
72  "price": "2999",
73  "promotionPrice": "1999",
74  "originalPrice": "3999",
75  "pic": "http://macro-oss.oss-cn-shenzhen.aliyuncs.com/mall/images/20180615/xiaomi.jpg",
76  "sale": 999,
77  "hasStock": true,
78  "salecount": 199,
79  "putawayDate": "2021-04-03",
80  "brandId": 6,
81  "brandName": "小米",
82  "brandImg": "http://macro-oss.oss-cn-shenzhen.aliyuncs.com/mall/images/20190129/1e34aef2a409119018a4c6258e39ecfb_22_222.png",
83  "categoryId": 19,
84  "categoryName": "手机通讯",
85  "attrs": [
86  {
87  "attrId": 1,
88  "attrName": "cpu",
89  "attrValue": "2核"
90  },
91  {
92  "attrId": 2,
93  "attrName": "颜色",
94  "attrValue": "蓝色"
95  }
96  ]
97  }
98  PUT /product_db/_doc/4
99  {
100  "id": "29",
101  "name": "Apple iPhone 8 Plus 64GB 金色特别版 移动联通电信4G手机",
102  "keywords": "苹果手机",
103  "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。",
104  "price": "5999",
105  "promotionPrice": "4999",
106  "originalPrice": "7999",
107  "pic": "http://macro-oss.oss-cn-shenzhen.aliyuncs.com/mall/images/20180615/5acc5248N6a5f81cd.jpg",
108  "sale": 999,
109  "hasStock": true,
110  "salecount": 1199,
111  "putawayDate": "2021-04-04",
112  "brandId": 51,
113  "brandName": "苹果",
114  "brandImg": "http://macro-oss.oss-cn-shenzhen.aliyuncs.com/mall/images/20180607/timg.jpg",
115  "categoryId": 19,
116  "categoryName": "手机通讯",
117  "attrs": [
118  {
119  "attrId": 1,
```

```
120   "attrName": "cpu",
121   "attrValue": "4核"
122 },
123 {
124   "attrId": 2,
125   "attrName": "颜色",
126   "attrValue": "金色"
127 }
128 ]
129 }
130 PUT /product_db/_doc/5
131 {
132   "id": "30",
133   "name": "HLA海澜之家简约动物印花短袖T恤",
134   "keywords": "海澜之家衣服",
135   "subTitle": "HLA海澜之家短袖T恤",
136   "price": "199",
137   "promotionPrice": "99",
138   "originalPrice": "299",
139   "pic": "http://macro-oss.oss-cn-shenzhen.aliyuncs.com/mall/images/20180615/5ad83a4fN6ff67ecd.jpg!cc_350x449.jpg",
140   "sale": 999,
141   "hasStock": true,
142   "salecount": 19,
143   "putawayDate": "2021-04-05",
144   "brandId": 50,
145   "brandName": "海澜之家",
146   "brandImg": "http://macro-oss.oss-cn-shenzhen.aliyuncs.com/mall/images/20190129/99d3279f1029d32b929343b09d3c72de_222_222.jpg",
147   "categoryId": 8,
148   "categoryName": "T恤",
149   "attrs": [
150   {
151     "attrId": 3,
152     "attrName": "尺寸",
153     "attrValue": "M"
154   },
155   {
156     "attrId": 4,
157     "attrName": "颜色",
158     "attrValue": "黑色"
159   }
160 ]
161 }
162 PUT /product_db/_doc/6
163 {
164   "id": "31",
165   "name": "HLA海澜之家蓝灰花纹圆领针织布短袖T恤",
166   "keywords": "海澜之家衣服",
167   "subTitle": "HLA海澜之家短袖T恤",
168   "price": "299",
169   "promotionPrice": "199",
170   "originalPrice": "299",
171   "pic": "http://macro-oss.oss-cn-shenzhen.aliyuncs.com/mall/images/20180615/5ac98b64N70acd82f.jpg!cc_350x449.jpg",
172   "sale": 999,
173   "hasStock": true,
174   "salecount": 399,
175   "putawayDate": "2021-04-06",
176   "brandId": 50,
177   "brandName": "海澜之家",
178   "brandImg": "http://macro-oss.oss-cn-shenzhen.aliyuncs.com/mall/images/20190129/99d3279f1029d32b929343b09d3c72de_222_222.jpg",
```

```
179 "categoryId": 8,
180 "categoryName": "T恤",
181 "attrs": [
182 {
183 "attrId": 3,
184 "attrName": "尺寸",
185 "attrValue": "X"
186 },
187 {
188 "attrId": 4,
189 "attrName": "颜色",
190 "attrValue": "蓝灰"
191 }
192 ]
193 }
194 PUT /product_db/_doc/7
195 {
196 "id": "32",
197 "name": "HLA海澜之家短袖T恤男基础款",
198 "keywords": "海澜之家衣服",
199 "subTitle": "HLA海澜之家短袖T恤",
200 "price": "269",
201 "promotionPrice": "169",
202 "originalPrice": "399",
203 "pic": "http://macro-oss.oss-cn-shenzhen.aliyuncs.com/mall/images/20180615/5a51eb88Na4797877.jpg",
204 "sale": 999,
205 "hasStock": true,
206 "salecount": 399,
207 "putawayDate": "2021-04-07",
208 "brandId": 50,
209 "brandName": "海澜之家",
210 "brandImg": "http://macro-oss.oss-cn-shenzhen.aliyuncs.com/mall/images/20190129/99d3279f1029d32b929343b09d3c72de_222_222.jpg",
211 "categoryId": 8,
212 "categoryName": "T恤",
213 "attrs": [
214 {
215 "attrId": 3,
216 "attrName": "尺寸",
217 "attrValue": "L"
218 },
219 {
220 "attrId": 4,
221 "attrName": "颜色",
222 "attrValue": "蓝色"
223 }
224 ]
225 }
226 PUT /product_db/_doc/8
227 {
228 "id": "33",
229 "name": "小米（MI）小米电视4A ",
230 "keywords": "小米电视机家用电器",
231 "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",
232 "price": "2269",
233 "promotionPrice": "2169",
234 "originalPrice": "2399",
235 "pic": "http://macro-oss.oss-cn-shenzhen.aliyuncs.com/mall/images/20180615/5b02804dN66004d73.jpg",
236 "sale": 999,
237 "hasStock": true,
```



```
238 "salecount":132,
239 "putawayDate":"2021-04-09",
240 "brandId": 6,
241 "brandName": "小米",
242 "brandImg": "http://macro-oss.oss-cn-shenzhen.aliyuncs.com/mall/images/20190129/1e34aef2a409119018a4c6258e39ecfb_222_222.png",
243 "categoryId": 35,
244 "categoryName": "手机数码",
245 "attrs": [
246 {
247 "attrId": 5,
248 "attrName": "屏幕尺寸",
249 "attrValue": "52"
250 },
251 {
252 "attrId": 6,
253 "attrName": "机身颜色",
254 "attrValue": "黑色"
255 }
256 ]
257 }
258 PUT /product_db/_doc/9
259 {
260 "id": "34",
261 "name": "小米（MI）小米电视4A 65英寸",
262 "keywords": "小米电视机家用电器",
263 "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",
264 "price": "3269",
265 "promotionPrice": "3169",
266 "originalPrice": "3399",
267 "pic": "http://macro-oss.oss-cn-shenzhen.aliyuncs.com/mall/images/20180615/5b028530N51eee7d4.jpg",
268 "sale": 999,
269 "hasStock": true,
270 "salecount":999,
271 "putawayDate":"2021-04-10",
272 "brandId": 6,
273 "brandName": "小米",
274 "brandImg": "http://macro-oss.oss-cn-shenzhen.aliyuncs.com/mall/images/20190129/1e34aef2a409119018a4c6258e39ecfb_222_222.png",
275 "categoryId": 35,
276 "categoryName": "手机数码",
277 "attrs": [
278 {
279 "attrId": 5,
280 "attrName": "屏幕尺寸",
281 "attrValue": "65"
282 },
283 {
284 "attrId": 6,
285 "attrName": "机身颜色",
286 "attrValue": "金色"
287 }
288 ]
289 }
290 PUT /product_db/_doc/10
291 {
292 "id": "35",
293 "name": "耐克NIKE 男子 休闲鞋 ROSHE RUN 运动鞋 511881-010黑色41码",
294 "keywords": "耐克运动鞋 鞋子",
295 "subTitle": "耐克NIKE 男子 休闲鞋 ROSHE RUN 运动鞋 511881-010黑色41码",
296 "price": "569",
```

```
297 "promotionPrice": "369",
298 "originalPrice": "899",
299 "pic": "http://macro-oss.oss-cn-shenzhen.aliyuncs.com/mall/images/20180615/5b235bb9Nf606460b.jpg",
300 "sale": 999,
301 "hasStock": true,
302 "salecount": 399,
303 "putawayDate": "2021-04-11",
304 "brandId": 58,
305 "brandName": "NIKE",
306 "brandImg": "http://macro-oss.oss-cn-shenzhen.aliyuncs.com/mall/images/20180615/timg (51).jpg",
307 "categoryId": 29,
308 "categoryName": "男鞋",
309 "attrs": [
310 {
311 "attrId": 7,
312 "attrName": "尺码",
313 "attrValue": "42"
314 },
315 {
316 "attrId": 8,
317 "attrName": "颜色",
318 "attrValue": "黑色"
319 }
320 ]
321 }
322 PUT /product_db/_doc/11
323 {
324 "id": "36",
325 "name": "耐克NIKE 男子 气垫 休闲鞋 AIR MAX 90 ESSENTIAL 运动鞋 AJ1285-101白色41码",
326 "keywords": "耐克运动鞋 鞋子",
327 "subTitle": "AIR MAX 90 ESSENTIAL 运动鞋 AJ1285-101白色",
328 "price": "769",
329 "promotionPrice": "469",
330 "originalPrice": "999",
331 "pic": "http://macro-oss.oss-cn-shenzhen.aliyuncs.com/mall/images/20180615/5b19403eN9f0b3cb8.jpg",
332 "sale": 999,
333 "hasStock": true,
334 "salecount": 499,
335 "putawayDate": "2021-04-13",
336 "brandId": 58,
337 "brandName": "NIKE",
338 "brandImg": "http://macro-oss.oss-cn-shenzhen.aliyuncs.com/mall/images/20180615/timg (51).jpg",
339 "categoryId": 29,
340 "categoryName": "男鞋",
341 "attrs": [
342 {
343 "attrId": 7,
344 "attrName": "尺码",
345 "attrValue": "44"
346 },
347 {
348 "attrId": 8,
349 "attrName": "颜色",
350 "attrValue": "白色"
351 }
352 ]
353 }
354 PUT /product_db/_doc/12
355 {
356 "id": "37",
```

```
357 "name": "(华为)HUAWEI MateBook X Pro 2019款 13.9英寸3K触控全面屏 轻薄笔记本",
358 "keywords": "轻薄笔记本华为 笔记本电脑",
359 "subTitle": "轻薄华为笔记本 电脑",
360 "price": "4769",
361 "promotionPrice": "4469",
362 "originalPrice": "4999",
363 "pic": "http://tuling-mall.oss-cn-shenzhen.aliyuncs.com/tulingmall/images/20200317/800_800_1555752016264mp.png",
364 "sale": 999,
365 "hasStock": true,
366 "salecount": 699,
367 "putawayDate": "2021-04-14",
368 "brandId": 3,
369 "brandName": "华为",
370 "brandImg": "http://macro-oss.oss-cn-shenzhen.aliyuncs.com/mall/images/20190129/17f2dd9756d9d333bee8e60ce8c03e4c_222_222.jpg",
371 "categoryId": 19,
372 "categoryName": "手机通讯",
373 "attrs": [
374 {
375 "attrId": 9,
376 "attrName": "容量",
377 "attrValue": "16G"
378 },
379 {
380 "attrId": 10,
381 "attrName": "网络",
382 "attrValue": "4G"
383 }
384 ]
385 }
386 PUT /product_db/_doc/13
387 {
388 "id": "38",
389 "name": "华为nova6se 手机 绮境森林 全网通 (8G+128G)",
390 "keywords": "轻薄笔记本华为 手机",
391 "subTitle": "华为nova6se 手机",
392 "price": "6769",
393 "promotionPrice": "6469",
394 "originalPrice": "6999",
395 "pic": "http://macro-oss.oss-cn-shenzhen.aliyuncs.com/mall/images/20180607/5ac1bf58Ndefaac16.jpg",
396 "sale": 999,
397 "hasStock": true,
398 "salecount": 899,
399 "putawayDate": "2021-04-15",
400 "brandId": 3,
401 "brandName": "华为",
402 "brandImg": "http://macro-oss.oss-cn-shenzhen.aliyuncs.com/mall/images/20190129/17f2dd9756d9d333bee8e60ce8c03e4c_222_222.jpg",
403 "categoryId": 19,
404 "categoryName": "手机通讯",
405 "attrs": [
406 {
407 "attrId": 9,
408 "attrName": "容量",
409 "attrValue": "64G"
410 },
411 {
412 "attrId": 10,
413 "attrName": "网络",
414 "attrValue": "5G"
415 }
```

```
416   ]
417 }
418 PUT /product_db/_doc/14
419 {
420   "id": "39",
421   "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",
422   "keywords": "手机膜",
423   "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",
424   "price": "29",
425   "promotionPrice": "39",
426   "originalPrice": "49",
427   "pic": "http://tuling-mall.oss-cn-shenzhen.aliyuncs.com/tulingmall/images/20200311/6df99dab78bb2014.jpg",
428   "sale": 999,
429   "hasStock": true,
430   "salecount": 799,
431   "putawayDate": "2021-04-16",
432   "brandId": 51,
433   "brandName": "苹果",
434   "brandImg": "http://tuling-mall.oss-cn-shenzhen.aliyuncs.com/tulingmall/images/20200311/2b84746650fc122d67749a876c453619.png",
435   "categoryId": 30,
436   "categoryName": "手机配件",
437   "attrs": [
438     {
439       "attrId": 11,
440       "attrName": "手机膜-材料",
441       "attrValue": "钢化"
442     },
443     {
444       "attrId": 12,
445       "attrName": "手机膜-颜色",
446       "attrValue": "白色"
447     }
448   ]
449 }
450
451 PUT /product_db/_doc/15
452 {
453   "id": "40",
454   "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",
455   "keywords": "七匹狼服装 衣服",
456   "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",
457   "price": "129",
458   "promotionPrice": "139",
459   "originalPrice": "149",
460   "pic": "http://tuling-mall.oss-cn-shenzhen.aliyuncs.com/tulingmall/images/20200311/19e846e727dff337.jpg",
461   "sale": 999,
462   "hasStock": true,
463   "salecount": 199,
464   "putawayDate": "2021-04-20",
465   "brandId": 49,
466   "brandName": "七匹狼",
467   "brandImg": "http://macro-oss.oss-cn-shenzhen.aliyuncs.com/mall/images/20190129/18d8bc3eb13533fab466d702a0d3fd1f40345bcd.jpg",
468   "categoryId": 8,
469   "categoryName": "T恤",
470   "attrs": [
471     {
472       "attrId": 3,
473       "attrName": "尺寸",
474       "attrValue": "M"
```

```
475 },
476 {
477   "attrId": 4,
478   "attrName": "颜色",
479   "attrValue": "白色"
480 }
481 ]
482 }
483 PUT /product_db/_doc/16
484 {
485   "id": "41",
486   "name": "华为P40 Pro手机",
487   "keywords": "华为手机",
488   "subTitle": "华为P40 Pro手机",
489   "price": "2129",
490   "promotionPrice": "2139",
491   "originalPrice": "2149",
492   "pic": "http://macro-oss.oss-cn-shenzhen.aliyuncs.com/mall/images/20180607/5ac1bf58Ndefaac16.jpg",
493   "sale": 999,
494   "hasStock": true,
495   "salecount":199,
496   "putawayDate":"2021-05-03",
497   "brandId": 3,
498   "brandName": "华为",
499   "brandImg": "http://macro-oss.oss-cn-shenzhen.aliyuncs.com/mall/images/20190129/17f2dd9756d9d333bee8e60ce8c03e4c_222_222.jpg",
500   "categoryId": 19,
501   "categoryName": "手机通讯",
502   "attrs": [
503     {
504       "attrId": 9,
505       "attrName": "容量",
506       "attrValue": "128G"
507     },
508     {
509       "attrId": 10,
510       "attrName": "网络",
511       "attrValue": "5G"
512     }
513   ]
514 }
515 PUT /product_db/_doc/17
516 {
517   "id": "42",
518   "name": "朵唯智能手机 4G全网通 老人学生双卡双待手机",
519   "keywords": "朵唯手机",
520   "subTitle": "朵唯手机后置双摄, 国产虎贲芯片! 优化散热结构! 轻薄机身! 朵唯4月特惠!",
521   "price": "3129",
522   "promotionPrice": "3139",
523   "originalPrice": "3249",
524   "pic": "http://macro-oss.oss-cn-shenzhen.aliyuncs.com/mall/images/20180615/xiaomi.jpg",
525   "sale": 999,
526   "hasStock": true,
527   "salecount":1199,
528   "putawayDate":"2021-06-01",
529   "brandId": 59,
530   "brandName": "朵唯",
531   "brandImg": "http://tuling-mall.oss-cn-shenzhen.aliyuncs.com/tulingmall/images/20200311/2b84746650fc122d67749a876c453619.png",
532   "categoryId": 19,
533   "categoryName": "手机通讯",
```

```
534   "attrs": [  
535     {  
536       "attrId": 9,  
537       "attrName": "容量",  
538       "attrValue": "32G"  
539     },  
540     {  
541       "attrId": 10,  
542       "attrName": "网络",  
543       "attrValue": "4G"  
544     }  
545   ]  
546 }  
547
```

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](#)

```
1  POST /product_db/_doc/_search  
2  {  
3    "from": 0,  
4    "size": 8,  
5    "query": {  
6      "bool": {  
7        "must": [  
8          {  
9            "match": {  
10             "name": {  
11               "query": "手机"  
12             }  
13           }  
14         ],  
15       },  
16       "filter": [  
17         {  
18           "term": {  
19             "hasStock": {  
20               "value": true  
21             }  
22           }  
23         },  
24         {  
25           "range": {  
26             "price": {  
27               "from": "1",  
28               "to": "5000"  
29             }  
30           }  
31         }  
32       ]  
33     },  
34     "sort": [  
35       {  
36         "salecount": {  
37           "order": "asc"  
38         }  
39       }  
40     ],  
41     "aggregations": {
```

```
42 "brand_agg": {
43   "terms": {
44     "field": "brandId",
45     "size": 50
46   },
47   "aggregations": {
48     "brand_name_agg": {
49       "terms": {
50         "field": "brandName"
51       }
52     },
53     "brand_img_agg": {
54       "terms": {
55         "field": "brandImg"
56       }
57     }
58   },
59   "category_agg": {
60     "terms": {
61       "field": "categoryId",
62       "size": 50,
63       "min_doc_count": 1
64     },
65     "aggregations": {
66       "category_name_agg": {
67         "terms": {
68           "field": "categoryName"
69         }
70       }
71     }
72   },
73   "attr_agg": {
74     "nested": {
75       "path": "attrs"
76     },
77     "aggregations": {
78       "attr_id_agg": {
79         "terms": {
80           "field": "attrs.attrId"
81         },
82         "aggregations": {
83           "attr_name_agg": {
84             "terms": {
85               "field": "attrs.attrName"
86             }
87           }
88         },
89       "attr_value_agg": {
90         "terms": {
91           "field": "attrs.attrValue"
92         }
93       }
94     }
95   },
96   "highlight": {
97     "pre_tags": [
98       "<b style='color:red'>"
99     ]
100   }
101 }
```

```
102   ],
103   "post_tags": [
104     "</b>"
105   ],
106   "fields": {
107     "name": {}
108   }
109 }
110 }
111
112
```

```
1 GET product_db/_search
2 {
3   "from": 0,
4   "size": 20,
5   "query": {
6     "bool": {
7       "must": [
8         {
9           "multi_match": {
10            "query": "手机",
11            "fields": [
12              "name",
13              "keywords",
14              "subTitle"
15            ]
16          }
17        }
18      ],
19      "filter": [
20        {
21          "term": {
22            "hasStock": "true"
23          }
24        },
25        {
26          "range": {
27            "price": {
28              "gte": 2000,
29              "lte": 5000
30            }
31          }
32        }
33      ]
34    },
35    "aggs": {
36      "brandId_aggs": {
37        "terms": {
38          "field": "brandId",
39          "size": 10
40        },
41      },
42      "aggs": {
43        "brandName_aggs": {
44          "terms": {
45            "field": "brandName"
46          }
47        },
48      },
49    }
50  }
51 }
```



```
48 "brandImg_aggs": {
49   "terms": {
50     "field": "brandImg"
51   }
52 }
53 }
54 },
55 "categoryId_aggs": {
56   "terms": {
57     "field": "categoryId",
58     "size": 10
59   },
60   "aggs": {
61     "categoryName_aggs": {
62       "terms": {
63         "field": "categoryName"
64       }
65     }
66   }
67 },
68 "attrs_aggs": {
69   "nested": {
70     "path": "attrs"
71   },
72   "aggs": {
73     "attrId_aggs": {
74       "terms": {
75         "field": "attrs.attrId"
76       },
77       "aggs": {
78         "attrName_aggs": {
79           "terms": {
80             "field": "attrs.attrName"
81           }
82         },
83         "attrValue_aggs": {
84           "terms": {
85             "field": "attrs.attrValue"
86           }
87         }
88       }
89     }
90   }
91 },
92 },
93 "sort": [
94   {
95     "salecount": {
96       "order": "desc"
97     }
98   }
99 ],
100 "highlight": {
101   "fields": {
102     "*": {}
103   }
104 }
105 }
```

1.5 Java代码实现商品搜索功能

1) 引入依赖

```
1 <dependency>
2 <groupId>org.springframework.boot</groupId>
3 <artifactId>spring-boot-starter-data-elasticsearch</artifactId>
4 </dependency>
```

2) 核心代码

```
1 @Qualifier("restHighLevelClient")
2 @Autowired
3 private RestHighLevelClient client;
4
5
6 @Override
7 public ESResponseResult search(ESRequestParam param) {
8
9     try {
10         //1、构建检索对象-封装请求相关参数信息
11         SearchRequest searchRequest = startBuildRequestParam(param);
12
13         //2、进行检索操作
14         SearchResponse response = client.search(searchRequest, RequestOptions.DEFAULT);
15
16         //3、分析响应数据,封装成指定的格式
17         ESResponseResult responseResult = startBuildResponseResult(response, param);
18         return responseResult;
19     } catch (Exception e) {
20         e.printStackTrace();
21     }
22
23     return null;
24
25 }
26
27 /**
28  * 封装请求参数信息
29  * 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
30  *
31  */
32 private SearchRequest startBuildRequestParam(ESRequestParam param) {
33
34     SearchSourceBuilder searchSourceBuilder = new SearchSourceBuilder();
35
36     /**
37      * 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
38      */
39     BoolQueryBuilder boolQueryBuilder = new BoolQueryBuilder();
40
41     //1、
42     if (!StringUtils.isEmpty(param.getKeyword())) {
43         //单字段查询
44         //boolQueryBuilder.must(QueryBuilders.matchQuery("name", param.getKeyword()));
45         //多字段查询
46         boolQueryBuilder.must(QueryBuilders.multiMatchQuery(param.getKeyword(), "name", "keywords", "subTitle"));
47     }
48
49     //2、根据类目ID进行过滤
50     if (null != param.getCategoryId()) {
51         boolQueryBuilder.filter(QueryBuilders.termQuery("categoryId", param.getCategoryId()));
52     }
```

```

53
54 //3、根据品牌ID进行过滤
55 if (null != param.getBrandId() && param.getBrandId().size() > 0) {
56     boolQueryBuilder.filter(QueryBuilders.termsQuery("brandId", param.getBrandId()));
57 }
58
59 //4、根据属性进行相关过滤
60 if (param.getAttrs() != null && param.getAttrs().size() > 0) {
61
62     param.getAttrs().forEach(item -> {
63         //attrs=1_白色&2_4核
64         BoolQueryBuilder boolQuery = QueryBuilders.boolQuery();
65
66         //attrs=1_64G
67         String[] s = item.split("_");
68         String attrId = s[0];
69         String[] attrValues = s[1].split(":");//这个属性检索用的值
70         boolQuery.must(QueryBuilders.termQuery("attrs.attrId", attrId));
71         boolQuery.must(QueryBuilders.termsQuery("attrs.attrValue", attrValues));
72
73         NestedQueryBuilder nestedQueryBuilder = QueryBuilders.nestedQuery("attrs", boolQuery, ScoreMode.None);
74         boolQueryBuilder.filter(nestedQueryBuilder);
75     });
76
77 }
78
79 //5、是否有库存
80 if (null != param.getHasStock()) {
81     boolQueryBuilder.filter(QueryBuilders.termQuery("hasStock", param.getHasStock() == 1));
82 }
83
84 /* if (price.length == 2) {
85     //price: _5000
86     if (param.getPrice().startsWith("_")) {
87         rangeQueryBuilder.lte(price[1]);
88     }
89     else{
90         //price: 1_5000
91         rangeQueryBuilder.gte(price[0]).lte(price[1]);
92     }
93 }
94 } else if (price.length == 1) {
95     //price: 1_
96     if (param.getPrice().endsWith("_")) {
97         rangeQueryBuilder.gte(price[0]);
98     }
99 }*/
100
101 //6、根据价格过滤
102 if (!StringUtils.isEmpty(param.getPrice())) {
103     //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）
104     RangeQueryBuilder rangeQueryBuilder = QueryBuilders.rangeQuery("price");
105     String[] price = param.getPrice().split("_");
106     if (price.length == 2) {
107         rangeQueryBuilder.gte(price[0]).lte(price[1]);
108     } else if (price.length == 1) {
109         if (param.getPrice().startsWith("_")) {
110             rangeQueryBuilder.lte(price[1]);
111         }
112         if (param.getPrice().endsWith("_")) {
113             rangeQueryBuilder.gte(price[0]);
114         }
115     }
116 }

```

```

114 }
115 }
116 boolQueryBuilder.filter(rangeQueryBuilder);
117 }
118
119 //封装所有查询条件
120 searchSourceBuilder.query(boolQueryBuilder);
121
122
123 /**
124  * 实现排序、高亮、分页操作
125  */
126
127 //排序
128 //页面传入的参数值形式 sort=price_asc/desc
129 if (!StringUtils.isEmpty(param.getSort())) {
130     String sort = param.getSort();
131     String[] sortFiledIds = sort.split("_");
132     System.out.println("sortFiledIds:"+sortFiledIds.length);
133     if(!StringUtils.isEmpty(sortFiledIds[0])){
134         SortOrder sortOrder = "asc".equalsIgnoreCase(sortFiledIds[1]) ? SortOrder.ASC : SortOrder.DESC;
135         searchSourceBuilder.sort(sortFiledIds[0], sortOrder);
136     }
137 }
138
139 //分页查询
140 searchSourceBuilder.from((param.getPageNum() - 1) * SearchConstant.PAGE_SIZE);
141 searchSourceBuilder.size(SearchConstant.PAGE_SIZE);
142
143 //高亮显示
144 if (!StringUtils.isEmpty(param.getKeyword())) {
145
146     HighlightBuilder highlightBuilder = new HighlightBuilder();
147     highlightBuilder.field("name");
148     highlightBuilder.preTags("<b style='color:red'>");
149     highlightBuilder.postTags("</b>");
150
151     searchSourceBuilder.highlighter(highlightBuilder);
152 }
153
154
155 /**
156  * 对品牌、分类信息、属性信息进行聚合分析
157  */
158 //1. 按照品牌进行聚合
159 TermsAggregationBuilder brand_agg = AggregationBuilders.terms("brand_agg");
160 brand_agg.field("brandId").size(50);
161
162
163 //1.1 品牌的子聚合-品牌名聚合
164 brand_agg.subAggregation(AggregationBuilders.terms("brand_name_agg")
165     .field("brandName").size(1));
166 //1.2 品牌的子聚合-品牌图片聚合
167 brand_agg.subAggregation(AggregationBuilders.terms("brand_img_agg")
168     .field("brandImg").size(1));
169
170 searchSourceBuilder.aggregation(brand_agg);
171
172 //2. 按照分类信息进行聚合
173 TermsAggregationBuilder category_agg = AggregationBuilders.terms("category_agg");
174 category_agg.field("categoryId").size(50);

```

```

175
176 category_agg.subAggregation(AggregationBuilders.terms("category_name_agg").field("categoryName").size(1));
177
178 searchSourceBuilder.aggregation(category_agg);
179
180 //2. 按照属性信息进行聚合
181 NestedAggregationBuilder attr_agg = AggregationBuilders.nested("attr_agg", "attrs");
182 //2.1 按照属性ID进行聚合
183 TermsAggregationBuilder attr_id_agg = AggregationBuilders.terms("attr_id_agg").field("attrs.attrId");
184 attr_agg.subAggregation(attr_id_agg);
185 //2.1.1 在每个属性ID下, 按照属性名进行聚合
186 attr_id_agg.subAggregation(AggregationBuilders.terms("attr_name_agg").field("attrs.attrName").size(1));
187 //2.1.1 在每个属性ID下, 按照属性值进行聚合
188 attr_id_agg.subAggregation(AggregationBuilders.terms("attr_value_agg").field("attrs.attrValue").size(50));
189 searchSourceBuilder.aggregation(attr_agg);
190
191 System.out.println("构建的DSL语句 {}:" + searchSourceBuilder.toString());
192
193 SearchRequest searchRequest = new SearchRequest(new String[]{SearchConstant.INDEX_NAME}, searchSourceBuilder);
194
195 return searchRequest;
196 }
197
198
199 /**
200  * 封装查询到的结果信息
201  * 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
202  */
203 private ESResponseResult startBuildResponseResult(SearchResponse response, ESRequestParam param) {
204
205     ESResponseResult result = new ESResponseResult();
206
207     //1、获取查询到的商品信息
208     SearchHits hits = response.getHits();
209
210     List<EsProduct> esModels = new ArrayList<>();
211     //2、遍历所有商品信息
212     if (hits.getHits() != null && hits.getHits().length > 0) {
213         for (SearchHit hit : hits.getHits()) {
214             String sourceAsString = hit.getSourceAsString();
215             EsProduct esModel = JSON.parseObject(sourceAsString, EsProduct.class);
216
217             //2.1 判断是否按关键字检索, 若是就显示高亮, 否则不显示
218             if (!StringUtils.isEmpty(param.getKeyword())) {
219                 //2.2 拿到高亮信息显示标题
220                 HighlightField name = hit.getHighlightFields().get("name");
221                 //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)
222                 String nameValue = name != null ? name.getFragments()[0].string() : esModel.getName();
223                 esModel.setName(nameValue);
224             }
225             esModels.add(esModel);
226         }
227     }
228     result.setProducts(esModels);
229
230     //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
231     Set<ESResponseResult.BrandVo> brandVos = new LinkedHashSet<>();
232     //获取到品牌的聚合
233     ParsedLongTerms brandAgg = response.getAggregations().get("brand_agg");
234     for (Terms.Bucket bucket : brandAgg.getBuckets()) {
235         ESResponseResult.BrandVo brandVo = new ESResponseResult.BrandVo();

```

```
236
237 //获取品牌的id
238 long brandId = bucket.getKeyAsNumber().longValue();
239 brandVo.setBrandId(brandId);
240
241 //获取品牌的名字
242 ParsedStringTerms brandNameAgg = bucket.getAggregations().get("brand_name_agg");
243 String brandName = brandNameAgg.getBuckets().get(0).getKeyAsString();
244 brandVo.setBrandName(brandName);
245
246 //获取品牌的LOGO
247 ParsedStringTerms brandImgAgg = bucket.getAggregations().get("brand_img_agg");
248 String brandImg = brandImgAgg.getBuckets().get(0).getKeyAsString();
249 brandVo.setBrandImg(brandImg);
250 System.out.println("brandId:"+brandId+"brandName:"+brandName+"brandImg");
251 brandVos.add(brandVo);
252 }
253 System.out.println("brandVos.size:"+brandVos.size());
254 result.setBrands(brandVos);
255
256
257 //4、当前商品相关的所有类目信息
258 //获取到分类的聚合
259 List<ESResponseResult.categoryVo> categoryVos = new ArrayList<>();
260
261 ParsedLongTerms categoryAgg = response.getAggregations().get("category_agg");
262
263
264 for (Terms.Bucket bucket : categoryAgg.getBuckets()) {
265     ESResponseResult.categoryVo categoryVo = new ESResponseResult.categoryVo();
266     //获取分类id
267     String keyAsString = bucket.getKeyAsString();
268     categoryVo.setCategoryId(Long.parseLong(keyAsString));
269
270     //获取分类名
271     ParsedStringTerms categoryNameAgg = bucket.getAggregations().get("category_name_agg");
272     String categoryName = categoryNameAgg.getBuckets().get(0).getKeyAsString();
273     categoryVo.setCategoryName(categoryName);
274     categoryVos.add(categoryVo);
275 }
276
277 result.setCategorys(categoryVos);
278
279
280
281 //5、获取商品相关的所有属性信息
282 List<ESResponseResult.AttrVo> attrVos = new ArrayList<>();
283 //获取属性信息的聚合
284 ParsedNested attrsAgg = response.getAggregations().get("attr_agg");
285 ParsedLongTerms attrIdAgg = attrsAgg.getAggregations().get("attr_id_agg");
286 for (Terms.Bucket bucket : attrIdAgg.getBuckets()) {
287     ESResponseResult.AttrVo attrVo = new ESResponseResult.AttrVo();
288     //获取属性ID值
289     long attrId = bucket.getKeyAsNumber().longValue();
290     attrVo.setAttrId(attrId);
291
292     //获取属性的名字
293     ParsedStringTerms attrNameAgg = bucket.getAggregations().get("attr_name_agg");
294     String attrName = attrNameAgg.getBuckets().get(0).getKeyAsString();
295     attrVo.setAttrName(attrName);
296
297 }
```

```

297 //获取属性的值
298 ParsedStringTerms attrValueAgg = bucket.getAggregations().get("attr_value_agg");
299 List<String> attrValues = attrValueAgg.getBuckets().stream().map(item -> item.getKeyAsString()).collect(Collectors.toList());
300 attrVo.setAttrValue(attrValues);
301
302 attrVos.add(attrVo);
303 }
304
305 result.setAttrs(attrVos);
306
307 //6、进行分页操作
308 result.setPageNum(param.getPageNum());
309 //获取总记录数
310 long total = hits.getTotalHits().value;
311 result.setTotal(total);
312
313 //计算总页码
314 int totalPages = (int) total % SearchConstant.PAGE_SIZE == 0 ?
315 (int) total / SearchConstant.PAGE_SIZE : ((int) total / SearchConstant.PAGE_SIZE + 1);
316 result.setTotalPages(totalPages);
317
318 List<Integer> pageNavs = new ArrayList<>();
319 for (int i = 1; i <= totalPages; i++) {
320 pageNavs.add(i);
321 }
322 result.setPageNavs(pageNavs);
323
324 return result;
325 }
326

```

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

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：

```
1 npm install elasticsearch-dump -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

```
1 elasticdump --input=http://172.16.0.39:9200/companydatabase --output=http://172.16.0.20:9200/companydatabase --type=settings
2 elasticdump --input=http://172.16.0.39:9200/companydatabase --output=http://172.16.0.20:9200/companydatabase --type=mapping
3 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

```
1 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：

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

之后调用snapshot api创建repository：

```
1 curl -XPUT http://172.16.0.39:9200/_snapshot/my_backup -H 'Content-Type: application/json' -d '{
2   "type": "fs",
3   "settings": {
4     "location": "/usr/local/services/test"
5     "compress": true
6   }
7 }'
```

如果需从其它云厂商的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

```
1 curl -XPUT http://172.16.0.39:9200/_snapshot/my_s3_repository
2 {
3   "type": "s3",
4   "settings": {
5     "bucket": "my_bucket_name",
6     "region": "us-west"
7   }
8 }
```

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

```
1 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

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

6. 查看快照恢复状态

```
1 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

```
1 POST _reindex
2 {
3   "source": {
4     "remote": {
5       "host": "http://172.16.0.39:9200"
6     },
7     "index": "test1",
8     "query": {
9       "match": {
10        "title": "elasticsearch"
11      }
12    }
13  },
14  "dest": {
15    "index": "test2"
16  }
17 }
```

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：

```
1 input {
2   elasticsearch {
3     hosts => ["http://172.16.0.39:9200"]
4     index => "*"
5     docinfo => true
6   }
7 }
8 output {
9   elasticsearch {
10    hosts => ["http://172.16.0.20:9200"]
11    index => "%{[@metadata][_index]}"
12  }
13 }
```

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:

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

```

9     scroll => "50m"
10     docinfo => true
11     #会提取ES文档的元信息metadata
12 }
13 }
14 filter {
15     #Only matched data are send to output.
16 }
17 output {
18     #stdout {
19     #     codec => rubydebug
20     #}
21     elasticsearch {
22         #action => "index" #动作 固定值不需要修改
23         hosts => "10.88.8.5:9200" #ElasticSearch主机
24         index => "%{[@metadata][_index]}" #索引名称
25         document_type => "%{[@metadata][_type]}"
26         document_id => "%{[@metadata][_id]}"
27         user => "elastic" #用户名
28         password => "elastic" #密码
29     }
30 }

```

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。