Elasticsearch快速入门，掌握这些刚刚好！

记得刚接触Elasticsearch的时候，没找啥资料，直接看了遍Elasticsearch的中文官方文档，中文文档很久没更新了，一直都是2.3的版本。最近又重新看了遍6.0的官方文档，由于官方文档介绍的内容比较多，每次看都很费力，所以这次整理了其中最常用部分，写下了这篇入门教程，希望对大家有所帮助。

简介

Elasticsearch是一个基于Lucene的搜索服务器。它提供了一个分布式的全文搜索引擎，基于restful web接口。Elasticsearch是用Java语言开发的，基于Apache协议的开源项目，是目前最受欢迎的企业搜索引擎。Elasticsearch广泛运用于云计算中，能够达到实时搜索，具有稳定，可靠，快速的特点。

安装

Windows下的安装

Elasticsearch

- 下载Elasticsearch 6.2.2的zip包，并解压到指定目录，下载地址：**https://www.elastic.co/cn/downloads/past-releases/elasticsearch-6-2-2**

arch-6.2.2

查看

:台电脑 › 本地磁盘 (D:) › developer › env › elasticsearch-6.2.2

名称	修改日期	类型	大小
bin	2018/2/16 19:03	文件夹	
config	2019/3/30 12:38	文件夹	
data	2019/5/11 14:50	文件夹	
lib	2018/2/16 19:03	文件夹	
logs	2019/5/11 14:50	文件夹	
modules	2018/2/16 19:03	文件夹	
plugins	2019/5/11 14:49	文件夹	
LICENSE.txt	2018/2/16 18:57	文本文档	12 KB
NOTICE.txt	2018/2/16 19:02	文本文档	188 KB
README.textile	2018/2/16 18:57	TEXTILE 文件	10 KB

- 安装中文分词插件，在elasticsearch-6.2.2\bin目录下执行以下命令；

```
1    elasticsearch-plugin install https://github.com/medcl/elasticsearch-analysis-
     ik/releases/download/v6.2.2/elasticsearch-analysis-ik-6.2.2.zipCopy to
     clipboardErrorCopied
```

- 运行bin目录下的elasticsearch.bat启动Elasticsearch；

查看

台电脑 › 本地磁盘 (D:) › developer › env › elasticsearch-6.2.2 › bin

名称	修改日期	类型	大小
elasticsearch	2018/2/16 18:57	文件	2 KB
elasticsearch.bat	2018/2/16 18:57	Windows 批处理...	2 KB
elasticsearch-env	2018/2/16 18:57	文件	3 KB
elasticsearch-env.bat	2018/2/16 18:57	Windows 批处理...	2 KB
elasticsearch-keystore	2018/2/16 18:57	文件	1 KB
elasticsearch-keystore.bat	2018/2/16 18:57	Windows 批处理...	1 KB
elasticsearch-plugin	2018/2/16 18:57	文件	1 KB
elasticsearch-plugin.bat	2018/2/16 18:57	Windows 批处理...	1 KB
elasticsearch-service.bat	2018/2/16 18:57	Windows 批处理...	8 KB
elasticsearch-service-mgr.exe	2018/2/16 18:57	应用程序	102 KB
elasticsearch-service-x64.exe	2018/2/16 18:57	应用程序	102 KB
elasticsearch-translog	2018/2/16 18:57	文件	1 KB
elasticsearch-translog.bat	2018/2/16 18:57	Windows 批处理...	1 KB

Kibana

- 下载Kibana,作为访问Elasticsearch的客户端，请下载6.2.2版本的zip包，并解压到指定目录，下载地址：https://artifacts.elastic.co/downloads/kibana/kibana-6.2.2-windows-x86_64.zip

台电脑 › 本地磁盘 (D:) › developer › env › kibana-6.2.2-windows-x86_64 ›

名称	修改日期	类型	大小
bin	2018/2/17 3:20	文件夹	
config	2018/2/17 3:20	文件夹	
data	2019/3/30 16:45	文件夹	
node	2018/2/17 3:20	文件夹	
node_modules	2018/2/17 3:20	文件夹	
optimize	2018/2/17 3:20	文件夹	
plugins	2018/2/17 3:20	文件夹	
src	2018/2/17 3:20	文件夹	
ui_framework	2018/2/17 3:20	文件夹	
webpackShims	2018/2/17 3:20	文件夹	
LICENSE.txt	2018/2/17 3:20	文本文档	1 KB
NOTICE.txt	2018/2/17 3:20	文本文档	1,103 KB
package.json	2018/2/17 3:20	JSON 文件	1 KB
README.txt	2018/2/17 3:20	文本文档	5 KB

- 运行bin目录下的kibana.bat，启动Kibana的用户界面

台电脑 > 本地磁盘 (D:) > developer > env > kibana-6.2.2-windows-x86_64 > bin

名称	修改日期	类型	大小
kibana.bat	2018/2/17 3:20	Windows 批处理...	1 KB
kibana-keystore.bat	2018/2/17 3:20	Windows 批处理...	1 KB
kibana-plugin.bat	2018/2/17 3:20	Windows 批处理...	1 KB

- 访问 **http://localhost:5601** 即可打开Kibana的用户界面：
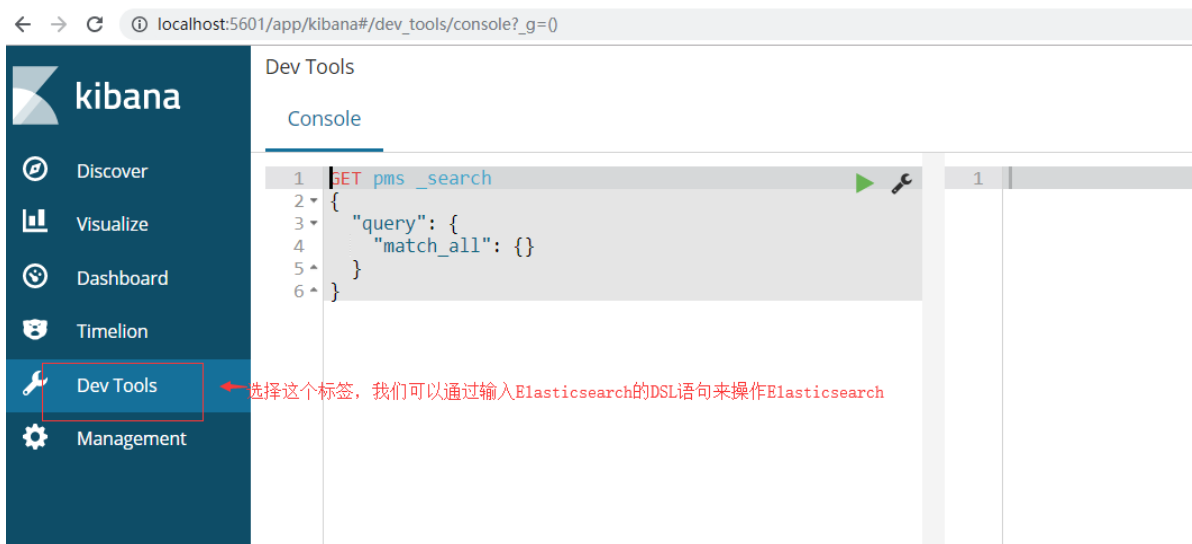
Linux**下的安装**

Elasticsearch

- 下载elasticsearch 6.4.0的docker镜像；

```
1    docker pull elasticsearch:6.4.0Copy to clipboardErrorCopied
```

- 修改虚拟内存区域大小，否则会因为过小而无法启动；

```
1    sysctl -w vm.max_map_count=262144Copy to clipboardErrorCopied
```

- 使用docker命令启动；

```
1    docker run -p 9200:9200 -p 9300:9300 --name elasticsearch \
2    -e "discovery.type=single-node" \
3    -e "cluster.name=elasticsearch" \
4    -v /mydata/elasticsearch/plugins:/usr/share/elasticsearch/plugins \
5    -v /mydata/elasticsearch/data:/usr/share/elasticsearch/data \
6    -d elasticsearch:6.4.0Copy to clipboardErrorCopied
```
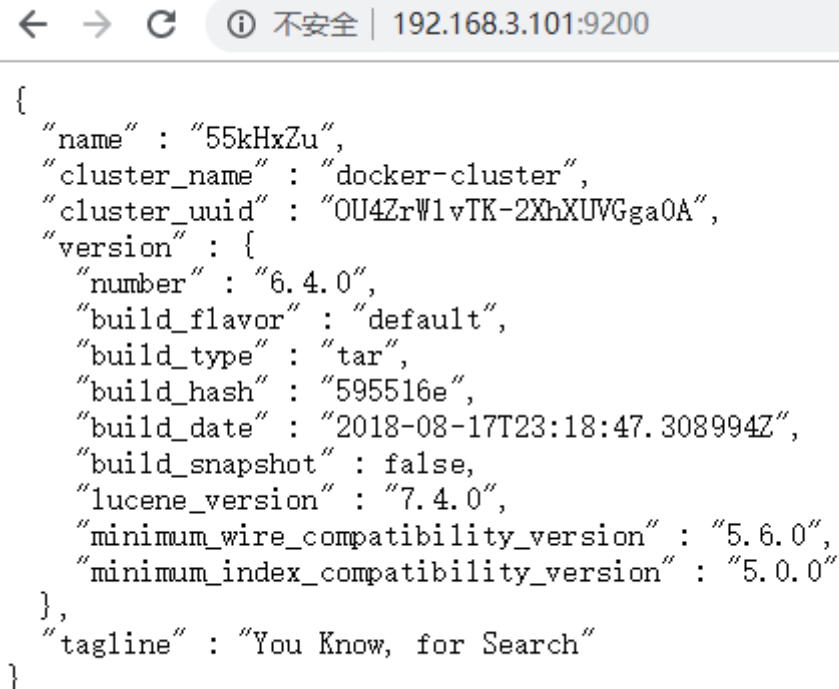
- 启动时会发现 /usr/share/elasticsearch/data 目录没有访问权限，只需要修改该目录的权限，再重新启动即可；

```
1    chmod 777 /mydata/elasticsearch/data/Copy to clipboardErrorCopied
```

- 安装中文分词器IKAnalyzer，并重新启动；

```
1    docker exec -it elasticsearch /bin/bash
2    #此命令需要在容器中运行
3    elasticsearch-plugin install https://github.com/medcl/elasticsearch-analysis-
     ik/releases/download/v6.4.0/elasticsearch-analysis-ik-6.4.0.zip
4    docker restart elasticsearchCopy to clipboardErrorCopied
```

- 访问会返回版本信息：**http://192.168.3.101:9200/**


```
{
  "name" : "55kHxZu",
  "cluster_name" : "docker-cluster",
  "cluster_uuid" : "OU4ZrW1vTK-2XhXUVGga0A",
  "version" : {
    "number" : "6.4.0",
    "build_flavor" : "default",
    "build_type" : "tar",
    "build_hash" : "595516e",
    "build_date" : "2018-08-17T23:18:47.308994Z",
    "build_snapshot" : false,
    "lucene_version" : "7.4.0",
    "minimum_wire_compatibility_version" : "5.6.0",
    "minimum_index_compatibility_version" : "5.0.0"
  },
  "tagline" : "You Know, for Search"
}
```
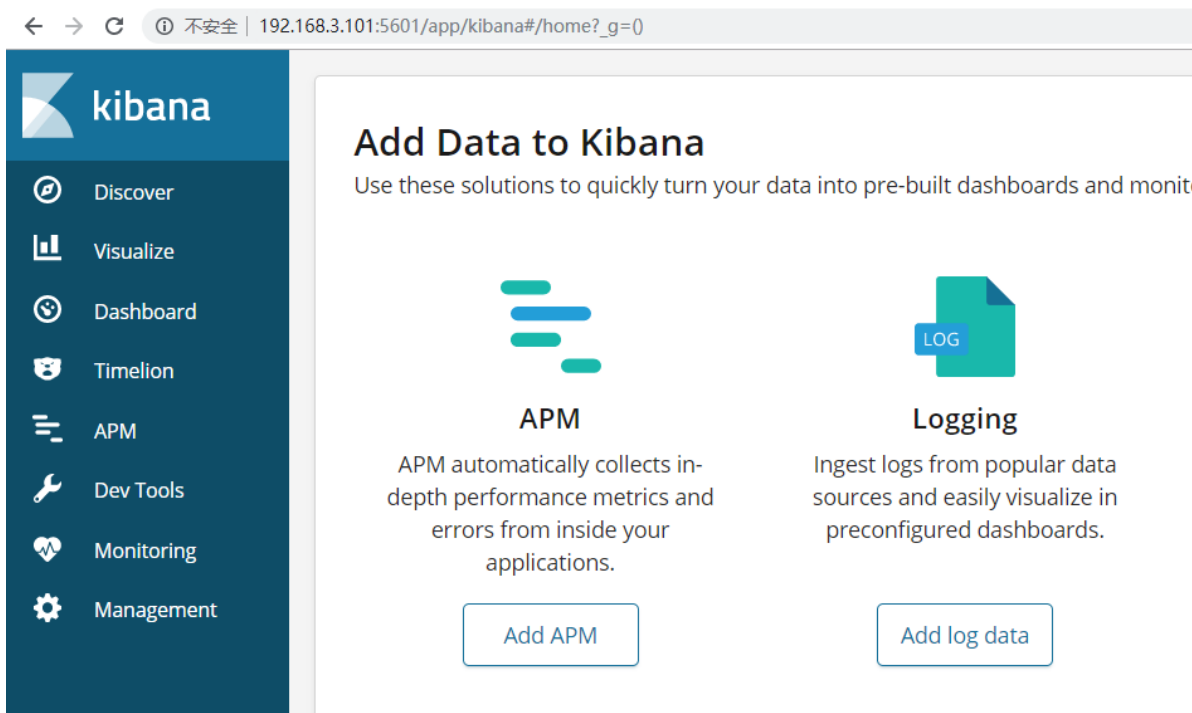
Kibina

- 下载kibana 6.4.0的docker镜像；

```
1    docker pull kibana:6.4.0Copy to clipboardErrorCopied
```

- 使用docker命令启动；

```
1    docker run --name kibana -p 5601:5601 \
2    --link elasticsearch:es \
3    -e "elasticsearch.hosts=http://es:9200" \
4    -d kibana:6.4.0Copy to clipboardErrorCopied
```

- 访问地址进行测试：**http://192.168.3.101:5601**

Add Data to Kibana

Use these solutions to quickly turn your data into pre-built dashboards and monito

APM

APM automatically collects in-depth performance metrics and errors from inside your applications.

Add APM

Logging

Ingest logs from popular data sources and easily visualize in preconfigured dashboards.

Add log data

相关概念

- Near Realtime（近实时）：Elasticsearch是一个近乎实时的搜索平台，这意味着从索引文档到可搜索文档之间只有一个轻微的延迟(通常是一秒钟)。
- Cluster（集群）：群集是一个或多个节点的集合，它们一起保存整个数据，并提供跨所有节点的联合索引和搜索功能。每个群集都有自己的唯一群集名称，节点通过名称加入群集。
- Node（节点）：节点是指属于集群的单个Elasticsearch实例，存储数据并参与集群的索引和搜索功能。可以将节点配置为按集群名称加入特定集群，默认情况下，每个节点都设置为加入一个名为 `elasticsearch` 的群集。
- Index（索引）：索引是一些具有相似特征的文档集合，类似于MySql中数据库的概念。
- Type（类型）：类型是索引的逻辑类别分区，通常，为具有一组公共字段的文档类型，类似MySql中表的概念。注意：在Elasticsearch 6.0.0及更高的版本中，一个索引只能包含一个类型。
- Document（文档）：文档是可被索引的基本信息单位，以JSON形式表示，类似于MySql中行记录的概念。
- Shards（分片）：当索引存储大量数据时，可能会超出单个节点的硬件限制，为了解决这个问题，Elasticsearch提供了将索引细分为分片的概念。分片机制赋予了索引水平扩容的能力、并允许跨分片分发和并行化操作，从而提高性能和吞吐量。
- Replicas（副本）：在可能出现故障的网络环境中，需要有一个故障切换机制，Elasticsearch提供了将索引的分片复制为一个或多个副本的功能，副本在某些节点失效的情况下提供高可用性。

集群状态查看

- 查看集群健康状态；

```
1   GET /_cat/health?vCopy to clipboardErrorCopied
2   epoch      timestamp cluster       status node.total node.data shards pri relo
    init unassign pending_tasks max_task_wait_time active_shards_percent
3   1585552862 15:21:02  elasticsearch yellow          1          1     27  27    0
       0      25            0                 -                 51.9%Copy to
    clipboardErrorCopied
```

- 查看节点状态；

```
1   GET /_cat/nodes?vCopy to clipboardErrorCopied
2   ip        heap.percent ram.percent cpu load_1m load_5m load_15m node.role master
    name
3   127.0.0.1           23          94  28                                mdi      *
    KFFjkpVCopy to clipboardErrorCopied
```

- 查看所有索引信息；

```
1   GET /_cat/indices?vCopy to clipboardErrorCopied
2   health status index    uuid                    pri rep docs.count docs.deleted
    store.size pri.store.size
3   green  open   pms      xlU0BjEoTrujDgeL6ENMPw    1   0         41            0
    30.5kb         30.5kb
4   green  open   .kibana  ljKQtJdwT9CnLrxbujdfWg    1   0          2            1
    10.7kb         10.7kbCopy to clipboardErrorCopied
```

索引操作

- 创建索引并查看；

```
1   PUT /customer
2   GET /_cat/indices?vCopy to clipboardErrorCopied
3   health status index    uuid                    pri rep docs.count docs.deleted
    store.size pri.store.size
4   yellow open   customer 9uPjf94gSq-SJS6eOuJrHQ    5   1          0            0
      460b           460b
5   green  open   pms      xlU0BjEoTrujDgeL6ENMPw    1   0         41            0
    30.5kb         30.5kb
6   green  open   .kibana  ljKQtJdwT9CnLrxbujdfWg    1   0          2            1
    10.7kb         10.7kbCopy to clipboardErrorCopied
```

- 删除索引并查看；

```
1   DELETE /customer
2   GET /_cat/indices?vCopy to clipboardErrorCopied
3   health status index    uuid                    pri rep docs.count docs.deleted
    store.size pri.store.size
4   green  open   pms      xlU0BjEoTrujDgeL6ENMPw    1   0         41            0
    30.5kb         30.5kb
5   green  open   .kibana  ljKQtJdwT9CnLrxbujdfWg    1   0          2            1
    10.7kb         10.7kbCopy to clipboardErrorCopied
```

类型操作

- 查看文档的类型；

```
1   GET /bank/account/_mappingCopy to clipboardErrorCopied
2   {
3     "bank": {
4       "mappings": {
5         "account": {
6           "properties": {
```

```
      7              "account_number": {
      8                "type": "long"
      9              },
     10              "address": {
     11                "type": "text",
     12                "fields": {
     13                  "keyword": {
     14                    "type": "keyword",
     15                    "ignore_above": 256
     16                  }
     17                }
     18              },
     19              "age": {
     20                "type": "long"
     21              },
     22              "balance": {
     23                "type": "long"
     24              },
     25              "city": {
     26                "type": "text",
     27                "fields": {
     28                  "keyword": {
     29                    "type": "keyword",
     30                    "ignore_above": 256
     31                  }
     32                }
     33              },
     34              "email": {
     35                "type": "text",
     36                "fields": {
     37                  "keyword": {
     38                    "type": "keyword",
     39                    "ignore_above": 256
     40                  }
     41                }
     42              },
     43              "employer": {
     44                "type": "text",
     45                "fields": {
     46                  "keyword": {
     47                    "type": "keyword",
     48                    "ignore_above": 256
     49                  }
     50                }
     51              },
     52              "firstname": {
     53                "type": "text",
     54                "fields": {
     55                  "keyword": {
     56                    "type": "keyword",
     57                    "ignore_above": 256
     58                  }
     59                }
     60              },
     61              "gender": {
     62                "type": "text",
     63                "fields": {
     64                  "keyword": {
```

```
65            "type": "keyword",
66            "ignore_above": 256
67          }
68        }
69      },
70      "lastname": {
71        "type": "text",
72        "fields": {
73          "keyword": {
74            "type": "keyword",
75            "ignore_above": 256
76          }
77        }
78      },
79      "state": {
80        "type": "text",
81        "fields": {
82          "keyword": {
83            "type": "keyword",
84            "ignore_above": 256
85          }
86        }
87      }
88    }
89   }
90  }
91  }
92 }Copy to clipboardErrorCopied
```

文档操作

- 在索引中添加文档;

```
1  PUT /customer/doc/1
2  {
3    "name": "John Doe"
4  }Copy to clipboardErrorCopied
5  {
6   "_index": "customer",
7   "_type": "doc",
8   "_id": "1",
9   "_version": 1,
10  "result": "created",
11  "_shards": {
12    "total": 2,
13    "successful": 1,
14    "failed": 0
15  },
16  "_seq_no": 3,
17  "_primary_term": 1
18 }Copy to clipboardErrorCopied
```

- 查看索引中的文档;

```
1    GET /customer/doc/1Copy to clipboardErrorCopied
2    {
3      "_index": "customer",
4      "_type": "doc",
5      "_id": "1",
6      "_version": 2,
7      "found": true,
8      "_source": {
9        "name": "John Doe"
10       }
11   }Copy to clipboardErrorCopied
```

- 修改索引中的文档:

```
1    POST /customer/doc/1/_update
2    {
3      "doc": { "name": "Jane Doe" }
4    }Copy to clipboardErrorCopied
5    {
6      "_index": "customer",
7      "_type": "doc",
8      "_id": "1",
9      "_version": 2,
10     "result": "updated",
11     "_shards": {
12       "total": 2,
13       "successful": 1,
14       "failed": 0
15     },
16     "_seq_no": 4,
17     "_primary_term": 1
18   }Copy to clipboardErrorCopied
```

- 删除索引中的文档;

```
1    DELETE /customer/doc/1Copy to clipboardErrorCopied
2    {
3      "_index": "customer",
4      "_type": "doc",
5      "_id": "1",
6      "_version": 3,
7      "result": "deleted",
8      "_shards": {
9        "total": 2,
10       "successful": 1,
11       "failed": 0
12     },
13     "_seq_no": 2,
14     "_primary_term": 1
15   }Copy to clipboardErrorCopied
```

- 对索引中的文档执行批量操作;

```
1    POST /customer/doc/_bulk
2    {"index":{"_id":"1"}}
3    {"name": "John Doe" }
4    {"index":{"_id":"2"}}
5    {"name": "Jane Doe" }Copy to clipboardErrorCopied
```

```
 6    {
 7      "took": 45,
 8      "errors": false,
 9      "items": [
10        {
11          "index": {
12            "_index": "customer",
13            "_type": "doc",
14            "_id": "1",
15            "_version": 3,
16            "result": "updated",
17            "_shards": {
18              "total": 2,
19              "successful": 1,
20              "failed": 0
21            },
22            "_seq_no": 5,
23            "_primary_term": 1,
24            "status": 200
25          }
26        },
27        {
28          "index": {
29            "_index": "customer",
30            "_type": "doc",
31            "_id": "2",
32            "_version": 1,
33            "result": "created",
34            "_shards": {
35              "total": 2,
36              "successful": 1,
37              "failed": 0
38            },
39            "_seq_no": 0,
40            "_primary_term": 1,
41            "status": 201
42          }
43        }
44      ]
45    }Copy to clipboardErrorCopied
```

数据搜索

> 查询表达式(Query DSL)是一种非常灵活又富有表现力的查询语言，Elasticsearch使用它可以以简单的JSON接口来实现丰富的搜索功能，下面的搜索操作都将使用它。

数据准备

- 首先我们需要导入一定量的数据用于搜索，使用的是银行账户表的例子，数据结构如下：

```
1    {
2        "account_number": 0,
3        "balance": 16623,
4        "firstname": "Bradshaw",
5        "lastname": "Mckenzie",
6        "age": 29,
7        "gender": "F",
8        "address": "244 Columbus Place",
9        "employer": "Euron",
10       "email": "bradshawmckenzie@euron.com",
11       "city": "Hobucken",
12       "state": "CO"
13   }Copy to clipboardErrorCopied
```

- 我们先复制下需要导入的数据，数据地址： **https://github.com/macrozheng/mall-learning/blob/master/document/json/accounts.json**
- 然后直接使用批量操作来导入数据，注意本文所有操作都在Kibana的Dev Tools中进行；

```
1    POST /bank/account/_bulk
2    {
3      "index": {
4        "_id": "1"
5      }
6    }
7    {
8      "account_number": 1,
9      "balance": 39225,
10     "firstname": "Amber",
11     "lastname": "Duke",
12     "age": 32,
13     "gender": "M",
14     "address": "880 Holmes Lane",
15     "employer": "Pyrami",
16     "email": "amberduke@pyrami.com",
17     "city": "Brogan",
18     "state": "IL"
19   }
20   ......省略若干条数据Copy to clipboardErrorCopied
```

- 导入完成后查看索引信息，可以发现 `bank` 索引中已经创建了1000条文档。

```
1  GET /_cat/indices?vCopy to clipboardErrorCopied
2  health status index   uuid                 pri rep docs.count docs.deleted
   store.size pri.store.size
3  yellow open   bank    HFjxDLNLRA-NATPKUQgjBw   5   1       1000              0
    474.6kb        474.6kbCopy to clipboardErrorCopied
```

搜索入门

- 最简单的搜索，使用 `match_all` 来表示，例如搜索全部；

```
1  GET /bank/_search
2  {
3    "query": { "match_all": {} }
4  }Copy to clipboardErrorCopied
```

```
{
  "took": 20,
  "timed_out": false,
  "_shards": {
    "total": 5,
    "successful": 5,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": 1000,
    "max_score": 1,
    "hits": [
      {
        "_index": "bank",
        "_type": "account",
        "_id": "25",
        "_score": 1,
        "_source": {
          "account_number": 25,
          "balance": 40540,
          "firstname": "Virginia",
          "lastname": "Ayala",
          "age": 39,
          "gender": "F",
          "address": "171 Putnam Avenue",
          "employer": "Filodyne",
          "email": "virginiaayala@filodyne.com",
          "city": "Nicholson",
          "state": "PA"
        }
      },
```

- 分页搜索，`from` 表示偏移量，从0开始，`size` 表示每页显示的数量；

```
1    GET /bank/_search
2    {
3      "query": { "match_all": {} },
4      "from": 0,
5      "size": 10
6    }Copy to clipboardErrorCopied
```

```
{
  "took": 2,
  "timed_out": false,
  "_shards": {
    "total": 5,
    "successful": 5,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": 1000,
    "max_score": 1,
    "hits": [
      {⟷},
      {⟷},
      {⟷},
      {⟷},
      {⟷},
      {⟷},
      {⟷},
      {⟷},
      {⟷},
      {⟷}
    ]
  }
}
```

- 搜索排序，使用 sort 表示，例如按 balance 字段降序排列；

```
1    GET /bank/_search
2    {
3      "query": { "match_all": {} },
4      "sort": { "balance": { "order": "desc" } }
5    }Copy to clipboardErrorCopied
```

```json
{
  "took": 63,
  "timed_out": false,
  "_shards": {
    "total": 5,
    "successful": 5,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": 1000,
    "max_score": null,
    "hits": [
      {
        "_index": "bank",
        "_type": "account",
        "_id": "248",
        "_score": null,
        "_source": {
          "account_number": 248,
          "balance": 49989,
          "firstname": "West",
          "lastname": "England",
          "age": 36,
          "gender": "M",
          "address": "717 Hendrickson Place",
          "employer": "Obliq",
          "email": "westengland@obliq.com",
          "city": "Maury",
          "state": "WA"
        },
        "sort": [
          49989
        ]
      },
      {
        "_index": "bank",
        "_type": "account",
        "_id": "854",
        "_score": null,
        "_source": {
          "account_number": 854,
          "balance": 49795,
          "firstname": "Jimenez",
          "lastname": "Barry",
          "age": 25,
          "gender": "F",
          "address": "603 Cooper Street",
```

- 搜索并返回指定字段内容，使用 `_source` 表示，例如只返回 `account_number` 和 `balance` 两个字段内容：

```
GET /bank/_search
{
  "query": { "match_all": {} },
  "_source": ["account_number", "balance"]
}Copy to clipboardErrorCopied
```

```
{
  "took": 23,
  "timed_out": false,
  "_shards": {
    "total": 5,
    "successful": 5,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": 1000,
    "max_score": 1,
    "hits": [
      {
        "_index": "bank",
        "_type": "account",
        "_id": "25",
        "_score": 1,
        "_source": {
          "account_number": 25,
          "balance": 40540
        }
      },
```

条件搜索

- 条件搜索，使用 `match` 表示匹配条件，例如搜索出 `account_number` 为 `20` 的文档：

```
1  GET /bank/_search
2  {
3    "query": {
4      "match": {
5        "account_number": 20
6      }
7    }
8  }Copy to clipboardErrorCopied
```

```json
{
  "took": 4,
  "timed_out": false,
  "_shards": {
    "total": 5,
    "successful": 5,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": 1,
    "max_score": 1,
    "hits": [
      {
        "_index": "bank",
        "_type": "account",
        "_id": "20",
        "_score": 1,
        "_source": {
          "account_number": 20,
          "balance": 16418,
          "firstname": "Elinor",
          "lastname": "Ratliff",
          "age": 36,
          "gender": "M",
          "address": "282 Kings Place",
          "employer": "Scentric",
          "email": "elinorratliff@scentric.com",
          "city": "Ribera",
          "state": "WA"
        }
      }
    ]
  }
}
```

- 文本类型字段的条件搜索，例如搜索 `address` 字段中包含 `mill` 的文档，对比上一条搜索可以发现，对于数值类型 `match` 操作使用的是精确匹配，对于文本类型使用的是模糊匹配;

```
1   GET /bank/_search
2   {
3     "query": {
4       "match": {
5         "address": "mill"
6       }
7     },
8     "_source": [
9       "address",
10      "account_number"
11     ]
12  }Copy to clipboardErrorCopied
```

```json
{
  "took": 5,
  "timed_out": false,
  "_shards": {
    "total": 5,
    "successful": 5,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": 4,
    "max_score": 4.89784,
    "hits": [
      {
        "_index": "bank",
        "_type": "account",
        "_id": "472",
        "_score": 4.89784,
        "_source": {
          "account_number": 472,
          "address": "288 Mill Street"
        }
      },
      {
        "_index": "bank",
        "_type": "account",
        "_id": "136",
        "_score": 4.8485627,
        "_source": {
          "account_number": 136,
          "address": "198 Mill Lane"
        }
      },
```

- 短语匹配搜索，使用 `match_phrase` 表示，例如搜索 `address` 字段中同时包含 `mill` 和 `lane` 的文档：

```
GET /bank/_search
{
  "query": {
    "match_phrase": {
      "address": "mill lane"
    }
  }
}Copy to clipboardErrorCopied
```

```
{
  "took": 21,
  "timed_out": false,
  "_shards": {
    "total": 5,
    "successful": 5,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": 1,
    "max_score": 8.398771,
    "hits": [
      {
        "_index": "bank",
        "_type": "account",
        "_id": "136",
        "_score": 8.398771,
        "_source": {
          "account_number": 136,
          "balance": 45801,
          "firstname": "Winnie",
          "lastname": "Holland",
          "age": 38,
          "gender": "M",
          "address": "198 Mill Lane",
          "employer": "Neteria",
          "email": "winnieholland@neteria.com",
          "city": "Urie",
          "state": "IL"
        }
      }
    ]
  }
}
```

组合搜索

- 组合搜索，使用 bool 来进行组合，must 表示同时满足，例如搜索 address 字段中同时包含 mill 和 lane 的文档；

```
1   GET /bank/_search
2   {
3     "query": {
4       "bool": {
5         "must": [
6           { "match": { "address": "mill" } },
7           { "match": { "address": "lane" } }
8         ]
9       }
10    }
11  }Copy to clipboardErrorCopied
```

```
{
  "took": 21,
  "timed_out": false,
  "_shards": {
    "total": 5,
    "successful": 5,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": 1,
    "max_score": 8.398771,
    "hits": [
      {
        "_index": "bank",
        "_type": "account",
        "_id": "136",
        "_score": 8.398771,
        "_source": {
          "account_number": 136,
          "balance": 45801,
          "firstname": "Winnie",
          "lastname": "Holland",
          "age": 38,
          "gender": "M",
          "address": "198 Mill Lane",
          "employer": "Neteria",
          "email": "winnieholland@neteria.com",
          "city": "Urie",
          "state": "IL"
        }
      }
    ]
  }
}
```

- 组合搜索，`should` 表示满足其中任意一个，搜索 `address` 字段中包含 `mill` 或者 `lane` 的文档；

```
1    GET /bank/_search
2    {
3      "query": {
4        "bool": {
5          "should": [
6            { "match": { "address": "mill" } },
7            { "match": { "address": "lane" } }
8          ]
9        }
10     }
11   }Copy to clipboardErrorCopied
```

```
{
  "took": 16,
  "timed_out": false,
  "_shards": {
    "total": 5,
    "successful": 5,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": 19,
    "max_score": 8.398771,
    "hits": [
      {
        "_index": "bank",
        "_type": "account",
        "_id": "136",
        "_score": 8.398771,
        "_source": {
          "account_number": 136,
          "balance": 45801,
          "firstname": "Winnie",
          "lastname": "Holland",
          "age": 38,
          "gender": "M",
          "address": "198 Mill Lane",
          "employer": "Neteria",
          "email": "winnieholland@neteria.com",
          "city": "Urie",
          "state": "IL"
        }
      },
      {
        "_index": "bank",
        "_type": "account",
        "_id": "472",
        "_score": 4.89784,
        "_source": {
          "account_number": 472,
          "balance": 25571,
          "firstname": "Lee",
          "lastname": "Long",
          "age": 32,
          "gender": "F",
          "address": "288 Mill Street",
          "employer": "Comverges",
          "email": "leelong@comverges.com",
```

- 组合搜索，`must_not` 表示同时不满足，例如搜索 `address` 字段中不包含 `mill` 且不包含 `lane` 的文档；

```
1    GET /bank/_search
2    {
3      "query": {
4        "bool": {
5          "must_not": [
6            { "match": { "address": "mill" } },
7            { "match": { "address": "lane" } }
8          ]
9        }
10     }
11   }Copy to clipboardErrorCopied
```

```
{
  "took": 14,
  "timed_out": false,
  "_shards": {
    "total": 5,
    "successful": 5,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": 981,
    "max_score": 1,
    "hits": [
      {
        "_index": "bank",
        "_type": "account",
        "_id": "25",
        "_score": 1,
        "_source": {
          "account_number": 25,
          "balance": 40540,
          "firstname": "Virginia",
          "lastname": "Ayala",
          "age": 39,
          "gender": "F",
          "address": "171 Putnam Avenue",
          "employer": "Filodyne",
          "email": "virginiaayala@filodyne.com",
          "city": "Nicholson",
          "state": "PA"
        }
      },
```

- 组合搜索，组合 `must` 和 `must_not`，例如搜索 `age` 字段等于 `40` 且 `state` 字段不包含 `ID` 的文档；

```
1   GET /bank/_search
2   {
3     "query": {
4       "bool": {
5         "must": [
6           { "match": { "age": "40" } }
7         ],
8         "must_not": [
9           { "match": { "state": "ID" } }
10        ]
11      }
12    }
13  }Copy to clipboardErrorCopied
```

```
{
  "took": 6,
  "timed_out": false,
  "_shards": {
    "total": 5,
    "successful": 5,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": 43,
    "max_score": 1,
    "hits": [
      {
        "_index": "bank",
        "_type": "account",
        "_id": "948",
        "_score": 1,
        "_source": {
          "account_number": 948,
          "balance": 37074,
          "firstname": "Sargent",
          "lastname": "Powers",
          "age": 40,
          "gender": "M",
          "address": "532 Fiske Place",
          "employer": "Accuprint",
          "email": "sargentpowers@accuprint.com",
          "city": "Umapine",
          "state": "AK"
        }
      },
```

过滤搜索

- 搜索过滤，使用 `filter` 来表示，例如过滤出 `balance` 字段在 `20000~30000` 的文档;

```
1   GET /bank/_search
2   {
3     "query": {
4       "bool": {
5         "must": { "match_all": {} },
6         "filter": {
```

```
 7                "range": {
 8                  "balance": {
 9                    "gte": 20000,
10                    "lte": 30000
11                  }
12                }
13              }
14            }
15          }
16      }Copy to clipboardErrorCopied
```

```
{
  "took": 32,
  "timed_out": false,
  "_shards": {
    "total": 5,
    "successful": 5,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": 217,
    "max_score": 1,
    "hits": [
      {
        "_index": "bank",
        "_type": "account",
        "_id": "253",
        "_score": 1,
        "_source": {
          "account_number": 253,
          "balance": 20240,
          "firstname": "Melissa",
          "lastname": "Gould",
          "age": 31,
          "gender": "M",
          "address": "440 Fuller Place",
          "employer": "Buzzopia",
          "email": "melissagould@buzzopia.com",
          "city": "Lumberton",
          "state": "MD"
        }
      },
      {
        "_index": "bank",
        "_type": "account",
        "_id": "400",
        "_score": 1,
        "_source": {
          "account_number": 400,
          "balance": 20685,
          "firstname": "Kane",
          "lastname": "King",
          "age": 21,
          "gender": "F",
          "address": "405 Cornelia Street",
          "employer": "Tri@Tribalog",
```

搜索聚合

- 对搜索结果进行聚合，使用 `aggs` 来表示，类似于MySql中的 `group by` ，例如对 `state` 字段进行聚合，统计出相同 `state` 的文档数量；

```
GET /bank/_search
{
  "size": 0,
  "aggs": {
    "group_by_state": {
      "terms": {
        "field": "state.keyword"
      }
    }
  }
}Copy to clipboardErrorCopied
```

```
"took": 96,
"timed_out": false,
"_shards": {
  "total": 5,
  "successful": 5,
  "skipped": 0,
  "failed": 0
},
"hits": {
  "total": 1000,
  "max_score": 0,
  "hits": []
},
"aggregations": {
  "group_by_state": {
    "doc_count_error_upper_bound": 20,
    "sum_other_doc_count": 770,
    "buckets": [
      {
        "key": "ID",
        "doc_count": 27
      },
      {
        "key": "TX",
        "doc_count": 27
      },
      {
        "key": "AL",
        "doc_count": 25
      },
      {
        "key": "MD",
        "doc_count": 25
      },
      {
        "key": "TN",
        "doc_count": 23
      },
```

- 嵌套聚合，例如对 `state` 字段进行聚合，统计出相同 `state` 的文档数量，再统计出 `balance` 的平均值；

```
GET /bank/_search
{
  "size": 0,
  "aggs": {
    "group_by_state": {
      "terms": {
        "field": "state.keyword"
      },
      "aggs": {
        "average_balance": {
          "avg": {
            "field": "balance"
          }
        }
      }
    }
  }
}Copy to clipboardErrorCopied
```

```
{
  "took": 11,
  "timed_out": false,
  "_shards": {
    "total": 5,
    "successful": 5,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": 1000,
    "max_score": 0,
    "hits": []
  },
  "aggregations": {
    "group_by_state": {
      "doc_count_error_upper_bound": 20,
      "sum_other_doc_count": 770,
      "buckets": [
        {
          "key": "ID",
          "doc_count": 27,
          "average_balance": {
            "value": 24368.777777777777
          }
        },
        {
          "key": "TX",
          "doc_count": 27,
          "average_balance": {
            "value": 27462.925925925927
          }
        },
```

- 对聚合搜索的结果进行排序，例如按 `balance` 的平均值降序排列；

```
1   GET /bank/_search
2   {
3     "size": 0,
4     "aggs": {
5       "group_by_state": {
6         "terms": {
7           "field": "state.keyword",
8           "order": {
9             "average_balance": "desc"
10          }
11        },
12        "aggs": {
13          "average_balance": {
14            "avg": {
15              "field": "balance"
16            }
17          }
18        }
19      }
20    }
21  }Copy to clipboardErrorCopied
```

```
  "max_score": 0,
  "hits": []
},
"aggregations": {
  "group_by_state": {
    "doc_count_error_upper_bound": -1,
    "sum_other_doc_count": 918,
    "buckets": [
      {
        "key": "AL",
        "doc_count": 6,
        "average_balance": {
          "value": 41418.166666666664
        }
      },
      {
        "key": "SC",
        "doc_count": 1,
        "average_balance": {
          "value": 40019
        }
      },
      {
        "key": "AZ",
        "doc_count": 10,
        "average_balance": {
          "value": 36847.4
        }
      },
      {
        "key": "VA",
        "doc_count": 13,
        "average_balance": {
          "value": 35418.846153846156
        }
      },
```

- 按字段值的范围进行分段聚合，例如分段范围为 `age` 字段的 `[20,30]` `[30,40]` `[40,50]`，之后按 `gender` 统计文档个数和 `balance` 的平均值；

```
GET /bank/_search
{
  "size": 0,
  "aggs": {
    "group_by_age": {
      "range": {
        "field": "age",
        "ranges": [
          {
            "from": 20,
            "to": 30
          },
          {
            "from": 30,
            "to": 40
          },
          {
            "from": 40,
            "to": 50
          }
        ]
      },
      "aggs": {
        "group_by_gender": {
          "terms": {
            "field": "gender.keyword"
          },
          "aggs": {
            "average_balance": {
              "avg": {
                "field": "balance"
              }
            }
          }
        }
      }
    }
  }
}Copy to clipboardErrorCopied
```

```json
"aggregations": {
  "group_by_age": {
    "buckets": [
      {
        "key": "20.0-30.0",
        "from": 20,
        "to": 30,
        "doc_count": 451,
        "group_by_gender": {
          "doc_count_error_upper_bound": 0,
          "sum_other_doc_count": 0,
          "buckets": [
            {
              "key": "M",
              "doc_count": 232,
              "average_balance": {
                "value": 27374.05172413793
              }
            },
            {
              "key": "F",
              "doc_count": 219,
              "average_balance": {
                "value": 25341.260273972603
              }
            }
          ]
        }
      },
      {
        "key": "30.0-40.0",
        "from": 30,
        "to": 40,
        "doc_count": 504,
        "group_by_gender": {
          "doc_count_error_upper_bound": 0,
          "sum_other_doc_count": 0,
          "buckets": [
            {
              "key": "F",
              "doc_count": 253,
              "average_balance": {
                "value": 25670.869565217392
              }
            },
```

参考资料

https://www.elastic.co/guide/en/elasticsearch/reference/6.0/getting-started.html