

主讲老师: Fox

有道云笔记地址: <https://note.youdao.com/s/17k3uiZJ>

1. Elasticsearch安装和简单配置

温馨提示: 初学者建议直接安装windows版本的ElasticSearch

安装文档: <https://www.elastic.co/guide/en/elasticsearch/reference/8.14/install-elasticsearch.html>

windows安装ElasticSearch

1) 下载ElasticSearch并解压

下载地址: https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-8.14.3-windows-x86_64.zip

ElasticSearch目录结构如下:

目录	描述
bin	脚本文件, 包括启动elasticsearch, 安装插件, 运行统计数据等
config	配置文件目录, 如elasticsearch配置、角色配置、jvm配置等。
jdk	7.x 以后特有, 自带的 java 环境
data	默认的数据存放目录, 包含节点、分片、索引、文档的所有数据, 生产环境需要修改。
lib	elasticsearch依赖的Java类库
logs	默认的日志文件存储路径, 生产环境需要修改。
modules	包含所有的Elasticsearch模块, 如Cluster、Discovery、Indices等。
plugins	已安装插件目录

2) 配置JDK环境

- ES比较耗内存, 建议虚拟机4G或以上内存, jvm1g以上的内存分配
- 运行Elasticsearch, 需安装并配置JDK。各个版本对Java的依赖 https://www.elastic.co/support/matrix#matrix_jvm
 - 7.0开始, 内置了Java环境。ES的JDK环境变量生效的优先级配置顺序ES_JAVA_HOME>ES_HOME
 - ES_JAVA_HOME: 这个环境变量用于指定Elasticsearch使用的Java运行时环境的路径。在启动Elasticsearch时, 它会检查ES_JAVA_HOME环境变量并使用其中的Java路径。

- ES_HOME：这个环境变量指定Elasticsearch的安装路径。它用于定位Elasticsearch的配置文件、插件和其他相关资源。设置ES_HOME环境变量可以让您在命令行中更方便地访问Elasticsearch的目录结构和文件。
- 可以参考ES的环境文件elasticsearch-env.bat

windows下，设置ES_JAVA_HOME和ES_HOME的环境变量

3) 配置ElasticSearch

编辑config/elasticsearch.yml 文件

关闭security安全认证

ES 8 默认是开启Security的，初学者便于快速上手，可以关闭Security。

编辑config/elasticsearch.yml 文件

4) 启动ElasticSearch服务

4.1) 解决启动日志乱码问题

```
1 #打开config/jvm.options 文件->末尾添加
2 -Dfile.encoding=GBK
```

4.2) 进入bin目录，点击elasticsearch.bat文件启动 ES 服务

注意：9300 端口为 Elasticsearch 集群间组件的通信端口，9200 端口为浏览器访问的 http 协议 RESTful 端口。

打开浏览器（推荐使用谷歌浏览器），输入地址：<http://localhost:9200>，测试结果

linux安装ElasticSearch

1) 环境准备

准备linux安装环境：

linux系统	IP	操作用户
centos7	192.168.65.47	fox

注意：ES不允许使用root账号启动服务，如果你当前账号是root，则需要创建一个专有账户

```
1 #为elasticsearch创建用户
2 adduser fox
3 passwd fox
```

2) 通过fox用户登录，下载ElasticSearch并解压

```
1 #centos7 通过fox用户进入
2 wget https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-8.14.3-linux-x86_64.tar.gz
3 tar -xzf elasticsearch-8.14.3-linux-x86_64.tar.gz
4 cd elasticsearch-8.14.3/
```

注意：如果在root用户下解压了ES安装包，可以通过下面的命令将ES安装包的所有者和组更改为fox用户

```
1 # 在root用户下操作
2 chown -R fox:fox elasticsearch-8.14.3
```

3) 配置JDK环境（可选）

```
1 # 进入fox用户主目录，比如/home/fox目录下，设置用户级别的环境变量
2 vim .bash_profile
3 #设置ES_JAVA_HOME和ES_HOME的路径
4 export ES_JAVA_HOME=/home/fox/elasticsearch-8.14.3/jdk/
5 export ES_HOME=/home/fox/elasticsearch-8.14.3
6 #执行以下命令使配置生效
7 source .bash_profile
```

4) 配置ElasticSearch

修改config/elasticsearch.yml配置文件

```
1 vim elasticsearch.yml
2
3 #配置节点对外提供服务的地址以及集群内通信的ip地址，默认为回环地址127.0.0.1 和[::1]
4 #配置为0.0.0.0开启远程访问支持
5 network.host: 0.0.0.0
6 #指定节点为单节点，可以绕过引导检查 初学者建议设置为此开发模式
7 discovery.type: single-node
8
9 #初学者建议关闭security安全认证
10 xpack.security.enabled: false
```

开发模式和生产模式

- **开发模式**：开发模式是默认配置（未配置集群发现设置），如果用户只是出于学习目的，而引导检查会把很多用户挡在门外，所以ES提供了一个设置项`discovery.type=single-node`。此项配置为指定节点为单节点，可以绕过引导检查。
- **生产模式**：当用户修改了有关集群的相关配置会触发生产模式，在生产模式下，服务启动会触发ES的引导检查或者叫启动检查（bootstrap checks），所谓引导检查就是在服务启动之前对一些重要的配置项进行检查，检查其配置值是否是合理的。引导检查包括对JVM大小、内存锁、虚拟内存、最大线程数、集群发现相关配置等相关的检查，如果某一项或者几项的配置不合理，ES会拒绝启动服务，并且在开发模式下的某些警告信息会升级成错误信息输出。引导检查十分严格，之所以宁可拒绝服务也要阻止用户启动服务是为了防止用户在对ES的基本使用不了解的前提下启动服务而导致的后期性能问题无法解决或者解决起来很麻烦。因为一旦服务以某种不合理的配置启动，时间久了之后可能会产生较大的性能问题，但此时集群已经变得难以维护和扩展，ES为了避免这种情况而做出了引导检查的设置，本来在开发模式下为警告的启动日志会升级为报错（Error）。这种设定虽然增加了用户的使用门槛，但是避免了日后产生更大的问题。

ElasticSearch常用配置参数

参考文档：<https://www.elastic.co/guide/en/elasticsearch/reference/8.14/important-settings.html>

- `cluster.name`

当前节点所属集群名称，多个节点如果要组成同一个集群，那么集群名称一定要配置成相同。默认值`elasticsearch`，生产环境建议根据ES集群的使用目的修改成合适的名字。不要在不同的环境中重用相同的集群名称，否则，节点可能会加入错误的集群。

- `node.name`

当前节点名称，默认值当前节点部署所在机器的主机名，所以如果一台机器上要起多个ES节点的话，需要通过配置该属性明确指定不同的节点名称。

- `path.data`

配置数据存储目录，比如索引数据等，默认值 `$ES_HOME/data`，生产环境下强烈建议部署到另外的安全目录，防止ES升级导致数据被误删除。

- `path.logs`

配置日志存储目录，比如运行日志和集群健康信息等，默认值 `$ES_HOME/logs`，生产环境下强烈建议部署到另外的安全目录，防止ES升级导致数据被误删除。

- `bootstrap.memory_lock`

配置ES启动时是否进行内存锁定检查，默认值`true`。

ES对于内存的需求比较大，一般生产环境建议配置大内存，如果内存不足，容易导致内存交换到磁盘，严重影响ES的性能。所以默认启动时进行相应大小内存的锁定，如果无法锁定则会启动失败。非生产环境可能机器内存本身就很小，能够供给ES使用的就更小，如果该参数配置为`true`的话很可能导致无法锁定内存以致ES无法成功启动，此时可以修改为`false`。

- `network.host`

节点对外提供服务的地址以及集群内通信的ip地址，默认值为当前节点所在机器的本机回环地址 `127.0.0.1` 和 `[::1]`，这就导致默认情况下只能通过当前节点所在主机访问当前节点。

- `http.port`

配置当前ES节点对外提供服务的http端口，默认 `9200`

- `transport.port`:

节点通信端口号，默认 `9300`

- `discovery.seed_hosts`

配置参与集群节点发现过程的主机列表，说白了就是集群中所有节点所在的主机列表，可以是具体的IP地址，也可以是可解析的域名。

- `cluster.initial_master_nodes`

配置ES集群初始化时参与master选举的节点名称列表，必须与`node.name`配置的一致。ES集群首次构建完成后，应该将集群中所有节点的配置文件中的`cluster.initial_master_nodes`配置项移除，重启集群或者将新节点加入某个已存在的集群时切记不要设置该配置项。

5) 配置JVM参数 (可选)

修改`config/jvm.options`配置文件，调整jvm堆内存大小

```
1 vim jvm.options
2 -Xms4g
3 -Xmx4g
```

配置的建议：

- `Xms` (JVM 启动时分配的最小堆内存) 和 `Xmx` (JVM 在运行过程中能够分配的最大堆内存) 设置成一样
- `Xmx`不要超过机器内存的50%

- 不要超过30GB - <https://www.elastic.co/cn/blog/a-heap-of-trouble>

6) 启动ElasticSearch服务

```
1 #注意: es默认不能用root用户启动
2 #fox用户下启动ES
3 bin/elasticsearch
4
5 # -d 后台启动
6 bin/elasticsearch -d
```

打开本地浏览器（推荐使用谷歌浏览器），输入地址：<http://192.168.65.47:9200>（换成linux环境对应的ip），测试结果如下：

生产模式启动ES服务常见错误总结

如果不配置discovery.type: single-node绕过引导检查，ES服务启动可能会抛出异常，比如提示如下：

[1]: max file descriptors [4096] for elasticsearch process is too low, increase to at least [65536]

ES因为需要大量的创建索引文件，需要大量的打开系统的文件，所以我们需要解除linux系统当中打开文件最大数目的限制，不然ES启动就会抛错

```
1 #切换到root用户
2 vim /etc/security/limits.conf
3
4 末尾添加如下配置:
5 *      soft    nofile    65536
6 *      hard    nofile    65536
7 *      soft    nproc     4096
8 *      hard    nproc     4096
```

[2]: max number of threads [1024] for user [es] is too low, increase to at least [4096]

无法创建本地线程问题,用户最大可创建线程数太小

```
1 vim /etc/security/limits.d/20-nproc.conf
2
3 改为如下配置:
4 * soft nproc 4096
```

[3]: max virtual memory areas vm.max_map_count [65530] is too low, increase to at least [262144]
最大虚拟内存太小,调大系统的虚拟内存

```
1 vim /etc/sysctl.conf
2 追加以下内容:
3 vm.max_map_count=262144
4 保存退出之后执行如下命令:
5 sysctl -p
```

[4]: the default discovery settings are unsuitable for production use; at least one of [discovery.seed_hosts, discovery.seed_providers, cluster.initial_master_nodes] must be configured
缺少默认配置, 至少需要配置discovery.seed_hosts/discovery.seed_providers、discovery.seed_providers、cluster.initial_master_nodes中的一个参数.

- discovery.seed_hosts: 集群主机列表
- discovery.seed_providers: 基于配置文件配置集群主机列表
- cluster.initial_master_nodes: 启动时初始化的参与选主的node, 生产环境必填

```
1 vim config/elasticsearch.yml
2 #添加配置
3 discovery.seed_hosts: ["127.0.0.1"]
4 cluster.initial_master_nodes: ["node-1"]
5
6 #或者指定配置单节点(开发模式 会绕过引导检查)
7 discovery.type: single-node
```

2. 安装ES浏览器插件

插件名称	插件图标	功能介绍	下载地址
Elasticsearch Head	image	方便查看集群节点数据 方便管理和索引、分片 支持同时连接多集群	Chrome下载 Github下载
Elasticsearch Tools	image-1677761829554	方便查看节点资源占用 可执行查询语句	Chrome下载
Elasticvue	image-1677761848792	功能强大对国人友好	Chrome下载 Edge下载

Elasticvue界面如下：

3. 可视化客户端Kibana安装

Kibana是一个开源分析和可视化平台，旨在与Elasticsearch协同工作。

参考文档：<https://www.elastic.co/guide/en/kibana/8.14/get-started.html>

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

1) 下载并解压缩Kibana

```
1 #windows
2 https://artifacts.elastic.co/downloads/kibana/kibana-8.14.3-windows-x86_64.zip
3 #linux
4 wget https://artifacts.elastic.co/downloads/kibana/kibana-8.14.3-linux-x86_64.tar.gz
5 tar -zxvf kibana-8.14.3-linux-x86_64.tar.gz
6 cd kibana-8.14.3
```

2) 修改Kibana.yml配置文件


```
1 vim config/kibana.yml
2
3 #指定Kibana服务器监听的端口号
4 server.port: 5601
5 #指定Kibana服务器绑定的主机地址
6 server.host: "0.0.0.0"
7 #指定Kibana连接到的Elasticsearch实例的访问地址
8 elasticsearch.hosts: ["http://localhost:9200"]
9 #将 Kibana 的界面语言设置为简体中文
10 i18n.locale: "zh-CN"
```

3) 运行Kibana

windows

直接执行kibana.bat

Linux

注意：kibana也需要非root用户启动

```
1 #启动kibana服务
2 bin/kibana
3 #后台启动，并将日志写入到logs/kibana.log
4 nohup bin/kibana > logs/kibana.log 2>&1 &
5
6 #查询kibana进程
7 netstat -tunlp | grep 5601
```

4) 访问Kibana: <http://localhost:5601>

cat API

```
1 /_cat/allocation          #查看单节点的shard分配整体情况
2 /_cat/shards              #查看各shard的详细情况
3 /_cat/shards/{index}     #查看指定分片的详细情况
4 /_cat/master              #查看master节点信息
5 /_cat/nodes               #查看所有节点信息
6 /_cat/indices             #查看集群中所有index的详细信息
7 /_cat/indices/{index}    #查看集群中指定index的详细信息
8 /_cat/segments           #查看各index的segment详细信息,包括segment名, 所属shard, 内存(磁盘)占
    用大小, 是否刷盘
9 /_cat/segments/{index}  #查看指定index的segment详细信息
10 /_cat/count              #查看当前集群的doc数量
11 /_cat/count/{index}     #查看指定索引的doc数量
12 /_cat/recovery           #查看集群内每个shard的recovery过程.调整replica。
13 /_cat/recovery/{index}  #查看指定索引shard的recovery过程
14 /_cat/health             #查看集群当前状态: 红、黄、绿
15 /_cat/pending_tasks     #查看当前集群的pending task
16 /_cat/aliases            #查看集群中所有alias信息,路由配置等
17 /_cat/aliases/{alias}   #查看指定索引的alias信息
18 /_cat/thread_pool        #查看集群各节点内部不同类型的threadpool的统计信息,
19 /_cat/plugins            #查看集群各个节点上的plugin信息
20 /_cat/fielddata          #查看当前集群各个节点的fielddata内存使用情况
21 /_cat/fielddata/{fields} #查看指定field的内存使用情况,里面传field属性对应的值
22 /_cat/nodeattrs          #查看单节点的自定义属性
23 /_cat/repositories       #输出集群中注册快照存储库
24 /_cat/templates          #输出当前正在存在的模板信息
```

4. 安装中文分词插件

Elasticsearch提供插件机制对系统进行扩展

在线安装

以安装analysis-icu这个分词插件为例

analysis-icu功能:

- 基于ICU (International Components for Unicode) 库, 提供高级的文本分析和处理功能。
- 支持多语言和复杂的Unicode文本处理。
- 包含ICU分词器 (ICU Tokenizer) 和ICU标准化过滤器 (ICU Normalizer) 。

analysis-icu应用场景:

- 多语言文本分析，适用于处理各种语言的文本。
- 支持Unicode标准化和处理复杂字符。
- 提供高级的文本处理功能，如正则表达式替换、文本转换等。

```
1 #查看已安装插件
2 bin/elasticsearch-plugin list
3 #安装插件
4 bin/elasticsearch-plugin install analysis-icu
5 #删除插件
6 bin/elasticsearch-plugin remove analysis-icu
```

注意：安装和删除完插件后，需要重启ES服务才能生效。

测试分词效果

```
1 POST _analyze
2 {
3   "analyzer":"icu_analyzer",
4   "text":"中华人民共和国"
5 }
```

离线安装

本地下载相应的插件，解压，然后手动上传到elasticsearch的plugins目录，然后重启ES实例就可以了。

比如ik中文分词插件：<https://github.com/medcl/elasticsearch-analysis-ik>

注意：ik分词器插件和ES版本必须一一对应，否则会出现兼容性问题导致ES启动失败。

当前ik分词器插件最新版本还只支持到ES8.4.1，而我们使用的ES版本是8.14.3，安装后会出现兼容性问题。那如何解决？

可以从<https://release.infinilabs.com/analysis-ik/stable/> 下载ES8.14.3对应版本的分词器

测试分词效果

```
1 #ES的默认分词设置是standard，会单字拆分
2 POST _analyze
3 {
4     "analyzer":"standard",
5     "text":"中华人民共和国"
6 }
7
8 #ik_smart:会做最粗粒度的拆
9 POST _analyze
10 {
11     "analyzer": "ik_smart",
12     "text": "中华人民共和国"
13 }
14
15 #ik_max_word:会将文本做最细粒度的拆分
16 POST _analyze
17 {
18     "analyzer":"ik_max_word",
19     "text":"中华人民共和国"
20 }
21
```

创建索引时可以指定IK分词器作为默认分词器

```
1 # 创建索引，指定默认分词器
2 PUT /employee
3 {
4     "settings" : {
5         "index" : {
6             "analysis.analyzer.default.type": "ik_max_word"
7         }
8     }
9 }
10
11 #查看索引setting信息
12 GET /employee/_settings
```

也可以针对字段配置IK分词器

```
1 #创建索引
2 PUT /index
3 # 指定content字段使用ik分词器
4 POST /index/_mapping
5 {
6   "properties": {
7     "content": {
8       "type": "text",
9       "analyzer": "ik_max_word",
10      "search_analyzer": "ik_smart"
11    }
12  }
13 }
14
15 #索引文档，也就是插入文档
16 POST /index/_create/1
17 {"content": "美国留给伊拉克的是个烂摊子吗"}
18
19 POST /index/_create/2
20 {"content": "公安部：各地校车将享最高路权"}
21
22 POST /index/_create/3
23 {"content": "中韩渔警冲突调查：韩警平均每天扣1艘中国渔船"}
24
25 POST /index/_create/4
26 {"content": "中国驻洛杉矶领事馆遭亚裔男子枪击 嫌犯已自首"}
27
28 #带高亮的查询
29 POST /index/_search
30 {
31   "query": {
32     "match": {
33       "content": "中国"
34     }
35   },
36   "highlight": {
37     "pre_tags": [
38       "<tag1>",
39       "<tag2>"
```

```
40 ],
41 "post_tags": [
42     "</tag1>",
43     "</tag2>"
44 ],
45 "fields": {
46     "content": {}
47 }
48 }
49 }
```

/index/_mapping 映射属性的解释：

- "properties": 这是一个包含字段定义的JSON对象。在这个例子中，它只包含了一个字段`content`。
- "content": 这是索引中要定义的字段名。
 - "type": "text": 指定`content`字段的数据类型为`text`。在Elasticsearch中，`text`类型用于全文搜索的文本字段，它可以被分词器（analyzer）处理成多个词条（tokens）用于索引和搜索。
 - "analyzer": "ik_max_word": 指定在索引（写入）`content`字段时使用的分词器为`ik_max_word`。
`ik_max_word`是Elasticsearch的IK分词器插件提供的一个分词器，它会对文本进行最细粒度的切分，以便尽可能多地捕获文本中的关键词，提高搜索的召回率。
 - "search_analyzer": "ik_smart": 指定在搜索（查询）`content`字段时使用的分词器为`ik_smart`。
`ik_smart`是IK分词器的另一种分词模式，它尝试对文本进行更智能的切分，以提高搜索的准确率。通过在索引和搜索时使用不同的分词器，可以在提高召回率的同时保持搜索的精度。