

安装elasticsearch

1. 部署单点es

1.1. 创建网络










因为我们还需要部署kibana容器，因此需要让es和kibana容器互联。这里先创建一个网络：

```
1 docker network create es-net
```

1.2. 加载镜像

这里我们采用elasticsearch的7.12.1版本的镜像，这个镜像体积非常大，接近1G。不建议大家自己pull。

课前资料提供了镜像的tar包：

	assets	文件夹
	hotel-demo	文件夹
	ik	文件夹
	static	文件夹
	elasticsearch-analysis-ik-7.12.1.zip	好压 ZIP 压缩文件
	es.tar	好压 TAR 压缩文件
	kibana.tar	好压 TAR 压缩文件
	tb_hotel.sql	SQL 源文件
	安装elasticsearch.md	Markdown File

大家将其上传到虚拟机中，然后运行命令加载即可：

```
1 # 导入数据
2 docker load -i es.tar
```

同理还有 **kibana** 的tar包也需要这样做。

1.3. 运行

运行docker命令，部署单点es：

```
1 docker run -d \  
2     --name es \  
3     -e "ES_JAVA_OPTS=-Xms512m -Xmx512m" \  
4     -e "discovery.type=single-node" \  
5     -v es-data:/usr/share/elasticsearch/data \  
6     -v es-plugins:/usr/share/elasticsearch/plugins \  
7     --privileged \  
8     --network es-net \  
9     -p 9200:9200 \  
10    -p 9300:9300 \  
11    elasticsearch:7.12.1
```

命令解释:

- `-e "cluster.name=es-docker-cluster"` : 设置集群名称
- `-e "http.host=0.0.0.0"` : 监听的地址, 可以外网访问
- `-e "ES_JAVA_OPTS=-Xms512m -Xmx512m"` : 内存大小
- `-e "discovery.type=single-node"` : 非集群模式
- `-v es-data:/usr/share/elasticsearch/data` : 挂载逻辑卷, 绑定es的数据目录
- `-v es-logs:/usr/share/elasticsearch/logs` : 挂载逻辑卷, 绑定es的日志目录
- `-v es-plugins:/usr/share/elasticsearch/plugins` : 挂载逻辑卷, 绑定es的插件目录
- `--privileged` : 授予逻辑卷访问权
- `--network es-net` : 加入一个名为es-net的网络中
- `-p 9200:9200` : 端口映射配置

在浏览器中输入: <http://192.168.150.101:9200> 即可看到elasticsearch的响应结果:

```
1  {
2    "name": "43deaa499668",
3    "cluster_name": "docker-cluster",
4    "cluster_uuid": "qBL32OsjQpeXHtrQsoJpaw",
5    "version": {
6      "number": "7.12.1",
7      "build_flavor": "default",
8      "build_type": "docker",
9      "build_hash": "3186837139b9c6b6d23c3200870651f10d3343b7",
10     "build_date": "2021-04-20T20:56:39.040728659Z",
11     "build_snapshot": false,
12     "lucene_version": "8.8.0",
13     "minimum_wire_compatibility_version": "6.8.0",
14     "minimum_index_compatibility_version": "6.0.0-beta1"
15   },
16   "tagline": "You Know, for Search"
17 }
```

2.部署kibana

kibana可以给我们提供一个elasticsearch的可视化界面，便于我们学习。

2.1.部署

运行docker命令，部署kibana

```
1  docker run -d \
2  --name kibana \
3  -e ELASTICSEARCH_HOSTS=http://es:9200 \
4  --network=es-net \
5  -p 5601:5601 \
6  kibana:7.12.1
```

- `--network es-net` : 加入一个名为es-net的网络中，与elasticsearch在同一个网络中
- `-e ELASTICSEARCH_HOSTS=http://es:9200` : 设置elasticsearch的地址，因为kibana已经与elasticsearch在一个网络，因此可以用容器名直接访问elasticsearch
- `-p 5601:5601` : 端口映射配置

kibana启动一般比较慢，需要多等待一会，可以通过命令：

```
1 docker logs -f kibana
```

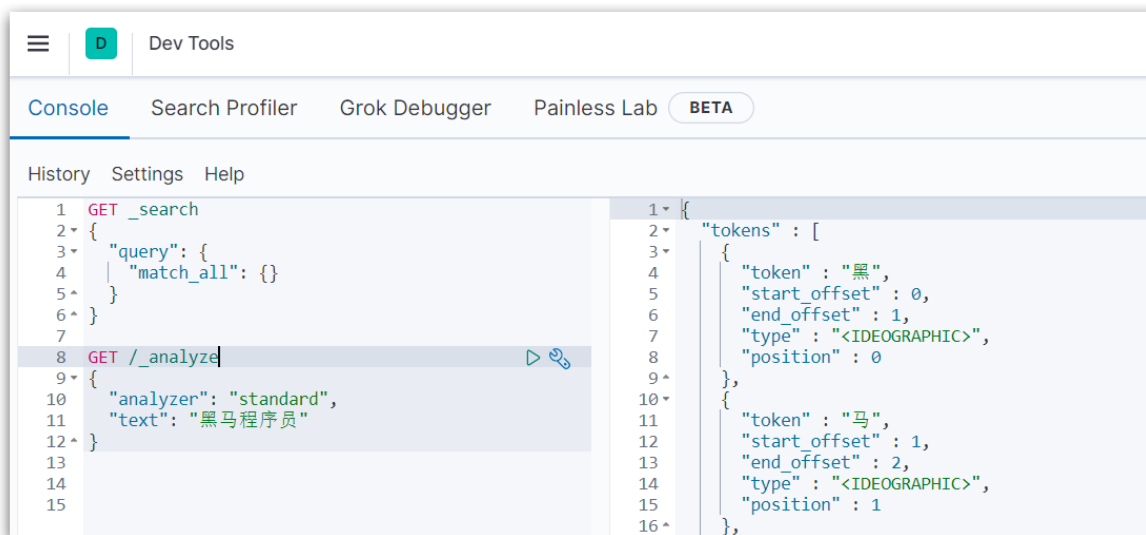
查看运行日志，当查看到下面的日志，说明成功：

```
6,"message":"Finished in 1121ms."}
{"type":"log","@timestamp":"2021-01-09T02:46:51Z","tags":["listening","info"],"pid":6,"message":"Server running at http://0:5601"}
{"type":"log","@timestamp":"2021-01-09T02:46:51Z","tags":["info","http","server","kibana"],"pid":6,"message":"http server running at http://0:5601"}
{"type":"log","@timestamp":"2021-01-09T02:46:51Z","tags":["status","plugin:spaces@7.2","info"],"pid":6,"state":"green","message":"Status changed from yellow to green Ready","prevState":"yellow","prevMsg":"Waiting for Elasticsearch"}
^C
```

此时，在浏览器输入地址访问：<http://192.168.150.101:5601>，即可看到结果

2.2.DevTools

kibana中提供了一个DevTools界面：



这个界面中可以编写DSL来操作elasticsearch。并且对DSL语句有自动补全功能。

3.安装IK分词器

3.1.在线安装ik插件（较慢）

```
1 # 进入容器内部
2 docker exec -it elasticsearch /bin/bash
3
4 # 在线下载并安装
5 ./bin/elasticsearch-plugin install
  https://github.com/medcl/elasticsearch-analysis-
  ik/releases/download/v7.12.1/elasticsearch-analysis-ik-7.12.1.zip
6
7 #退出
8 exit
9 #重启容器
10 docker restart elasticsearch
```

3.2. 离线安装ik插件（推荐）

1) 查看数据卷目录

安装插件需要知道elasticsearch的plugins目录位置，而我们用了数据卷挂载，因此需要查看elasticsearch的数据卷目录，通过下面命令查看：

```
1 docker volume inspect es-plugins
```

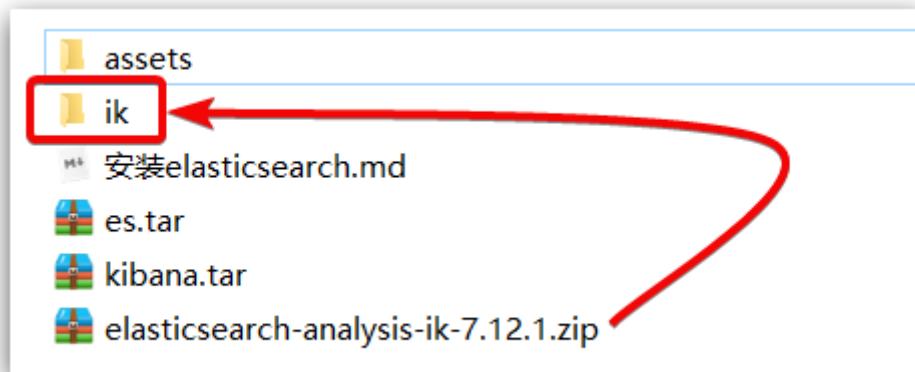
显示结果：

```
1 [
2   {
3     "CreatedAt": "2022-05-06T10:06:34+08:00",
4     "Driver": "local",
5     "Labels": null,
6     "Mountpoint": "/var/lib/docker/volumes/es-plugins/_data",
7     "Name": "es-plugins",
8     "Options": null,
9     "Scope": "local"
10  }
11 ]
```

说明plugins目录被挂载到了：`/var/lib/docker/volumes/es-plugins/_data` 这个目录中。

2) 解压缩分词器安装包

下面我们需要把课前资料中的ik分词器解压缩，重命名为ik



3) 上传到es容器的插件数据卷中

也就是 `/var/lib/docker/volumes/es-plugins/_data` :



4) 重启容器

```
1 # 4、重启容器
2 docker restart es
```

```
1 # 查看es日志
2 docker logs -f es
```

5) 测试:

IK分词器包含两种模式:

- `ik_smart` : 最少切分
- `ik_max_word` : 最细切分

```
1 GET /_analyze
2 {
3     "analyzer": "ik_max_word",
4     "text": "黑马程序员学习java太棒了"
5 }
```

结果:

```
1 {
2     "tokens" : [
3         {
4             "token" : "黑马",
5             "start_offset" : 0,
6             "end_offset" : 2,
7             "type" : "CN_WORD",
8             "position" : 0
9         },
10        {
11            "token" : "程序员",
12            "start_offset" : 2,
13            "end_offset" : 5,
14            "type" : "CN_WORD",
15            "position" : 1
16        },
17        {
18            "token" : "程序",
19            "start_offset" : 2,
20            "end_offset" : 4,
21            "type" : "CN_WORD",
22            "position" : 2
23        },
24        {
25            "token" : "员",
26            "start_offset" : 4,
27            "end_offset" : 5,
28            "type" : "CN_CHAR",
29            "position" : 3
30        },
31        {
32            "token" : "学习",
33            "start_offset" : 5,
34            "end_offset" : 7,
35            "type" : "CN_WORD",
36            "position" : 4
37        },
38        {
39            "token" : "java",
40            "start_offset" : 7,
41            "end_offset" : 11,
```

```

42     "type" : "ENGLISH",
43     "position" : 5
44 },
45 {
46     "token" : "太棒了",
47     "start_offset" : 11,
48     "end_offset" : 14,
49     "type" : "CN_WORD",
50     "position" : 6
51 },
52 {
53     "token" : "太棒",
54     "start_offset" : 11,
55     "end_offset" : 13,
56     "type" : "CN_WORD",
57     "position" : 7
58 },
59 {
60     "token" : "了",
61     "start_offset" : 13,
62     "end_offset" : 14,
63     "type" : "CN_CHAR",
64     "position" : 8
65 }
66 ]
67 }

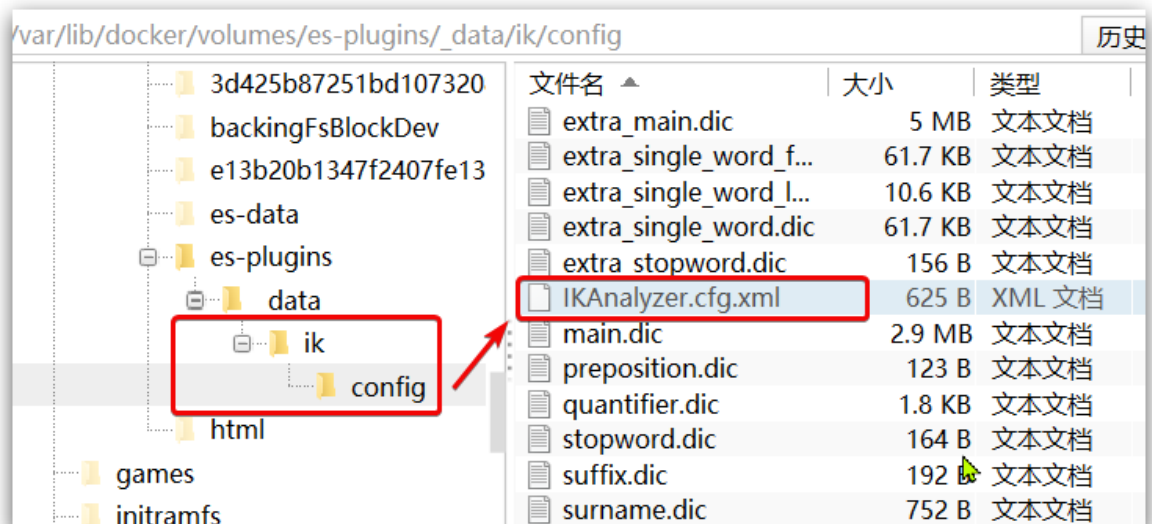
```

3.3 扩展词词典

随着互联网的发展，“造词运动”也越发的频繁。出现了很多新的词语，在原有的词汇列表中并不存在。比如：“奥力给”，“传智播客”等。

所以我们的词汇也需要不断的更新，IK分词器提供了扩展词汇的功能。

1) 打开IK分词器config目录：



2) 在IKAnalyzer.cfg.xml配置文件内容添加:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
3 <properties>
4     <comment>IK Analyzer 扩展配置</comment>
5     <!--用户可以在这里配置自己的扩展字典 *** 添加扩展词典-->
6     <entry key="ext_dict">ext.dic</entry>
7 </properties>
```

3) 新建一个 ext.dic, 可以参考config目录下复制一个配置文件进行修改

```
1 传智播客
2 奥力给
```

4) 重启elasticsearch

```
1 docker restart es
2
3 # 查看 日志
4 docker logs -f elasticsearch
```

```
{
  "type": "server",
  "timestamp": "2020-11-15T15:07:46,900Z",
  "level": "INFO",
  "component": "o.e.g.GatewayService",
  "cluster.name": "elastic-cluster",
  "node.name": "elasticsearch",
  "message": "recovered [5] indices into cluster state",
  "cluster.uid": "-F3RLuCiQUmothn55kpQVg",
  "node.id": "IEsNMhGqTH-qgDVyW_4cHA"
}
{"type": "server", "timestamp": "2020-11-15T15:07:47,162Z", "level": "INFO", "component": "o.w.a.d.Monitor", "cluster.name": "elastic-cluster", "node.name": "elasticsearch", "message": "try load config from /usr/share/elasticsearch/config/analysis-ik/IKAnalyzer.cfg.xml", "cluster.uid": "-F3RLuCiQUmothn55kpQVg", "node.id": "IEsNMhGqTH-qgDVyW_4cHA"}
{"type": "server", "timestamp": "2020-11-15T15:07:47,173Z", "level": "INFO", "component": "o.w.a.d.Monitor", "cluster.name": "elastic-cluster", "node.name": "elasticsearch", "message": "try load config from /usr/share/elasticsearch/plugins/ik/config/IKAnalyzer.cfg.xml", "cluster.uid": "-F3RLuCiQUmothn55kpQVg", "node.id": "IEsNMhGqTH-qgDVyW_4cHA"}
{"type": "server", "timestamp": "2020-11-15T15:07:47,796Z", "level": "INFO", "component": "o.w.a.d.Monitor", "cluster.name": "elastic-cluster", "node.name": "elasticsearch", "message": "[Dict Loading] /usr/share/elasticsearch/plugins/ik/config/ext.dic", "cluster.uid": "-F3RLuCiQUmothn55kpQVg", "node.id": "IEsNMhGqTH-qgDVyW_4cHA"}
{"type": "server", "timestamp": "2020-11-15T15:07:49,363Z", "level": "INFO", "component": "o.e.c.r.a.AllocationService", "cluster.name": "elastic-cluster", "node.name": "elasticsearch", "message": "Cluster health status changed from [RED] to [YELLOW] (reason: [shards started [[.kibana_1][0]]].", "cluster.uid": "-F3RLuCiQUmothn55kpQVg", "node.id": "IEsNMhGqTH-qgDVyW_4cHA" }
```

日志中已经成功加载ext.dic配置文件

5) 测试效果:

```
1 GET /_analyze
2 {
3     "analyzer": "ik_max_word",
4     "text": "传智播客Java就业超过90%,奥力给!"
5 }
```

注意当前文件的编码必须是 UTF-8 格式, 严禁使用Windows记事本编辑

3.4 停用词词典

在互联网项目中, 在网络间传输的速度很快, 所以很多语言是不允许在网络上传递的, 如: 关于宗教、政治等敏感词语, 那么我们在搜索时也应该忽略当前词汇。

IK分词器也提供了强大的停用词功能，让我们在索引时就直接忽略当前的停用词汇表中的内容。

1) IKAnalyzer.cfg.xml配置文件内容添加：

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
3 <properties>
4     <comment>IK Analyzer 扩展配置</comment>
5     <!--用户可以在这里配置自己的扩展字典-->
6     <entry key="ext_dict">ext.dic</entry>
7     <!--用户可以在这里配置自己的扩展停止词字典 *** 添加停用词
   词典-->
8     <entry key="ext_stopwords">stopword.dic</entry>
9 </properties>
```

3) 在 stopword.dic 添加停用词

```
1 习大大
```

4) 重启elasticsearch

```
1 # 重启服务
2 docker restart elasticsearch
3 docker restart kibana
4
5 # 查看 日志
6 docker logs -f elasticsearch
```

日志中已经成功加载stopword.dic配置文件

5) 测试效果：

```
1 GET /_analyze
2 {
3     "analyzer": "ik_max_word",
4     "text": "传智播客Java就业率超过95%,习大大都点赞,奥力给！"
5 }
```

注意当前文件的编码必须是 UTF-8 格式，严禁使用Windows记事本编辑

4.部署es集群

我们会在单机上利用docker容器运行多个es实例来模拟es集群。不过生产环境推荐大家每一台服务节点仅部署一个es的实例。

部署es集群可以直接使用docker-compose来完成，但这要求你的Linux虚拟机至少有4G的内存空间

4.1. 创建es集群

首先编写一个docker-compose文件，内容如下：

```
1 version: '2.2'
2 services:
3   es01:
4     image: elasticsearch:7.12.1
5     container_name: es01
6     environment:
7       - node.name=es01
8       - cluster.name=es-docker-cluster
9       - discovery.seed_hosts=es02,es03
10      - cluster.initial_master_nodes=es01,es02,es03
11      - "ES_JAVA_OPTS=-Xms512m -Xmx512m"
12     volumes:
13       - data01:/usr/share/elasticsearch/data
14     ports:
15       - 9200:9200
16     networks:
17       - elastic
18   es02:
19     image: elasticsearch:7.12.1
20     container_name: es02
21     environment:
22       - node.name=es02
23       - cluster.name=es-docker-cluster
24       - discovery.seed_hosts=es01,es03
25       - cluster.initial_master_nodes=es01,es02,es03
26       - "ES_JAVA_OPTS=-Xms512m -Xmx512m"
27     volumes:
28       - data02:/usr/share/elasticsearch/data
29     ports:
30       - 9201:9200
31     networks:
32       - elastic
33   es03:
34     image: elasticsearch:7.12.1
35     container_name: es03
36     environment:
37       - node.name=es03
38       - cluster.name=es-docker-cluster
39       - discovery.seed_hosts=es01,es02
40       - cluster.initial_master_nodes=es01,es02,es03
41       - "ES_JAVA_OPTS=-Xms512m -Xmx512m"
42     volumes:
```

```
43     - data03:/usr/share/elasticsearch/data
44     networks:
45     - elastic
46     ports:
47     - 9202:9200
48     volumes:
49     data01:
50         driver: local
51     data02:
52         driver: local
53     data03:
54         driver: local
55
56     networks:
57     elastic:
58         driver: bridge
```

es运行需要修改一些linux系统权限，修改 `/etc/sysctl.conf` 文件

```
1 vi /etc/sysctl.conf
```

添加下面的内容：

```
1 vm.max_map_count=262144
```

然后执行命令，让配置生效：

```
1 sysctl -p
```

通过docker-compose启动集群：

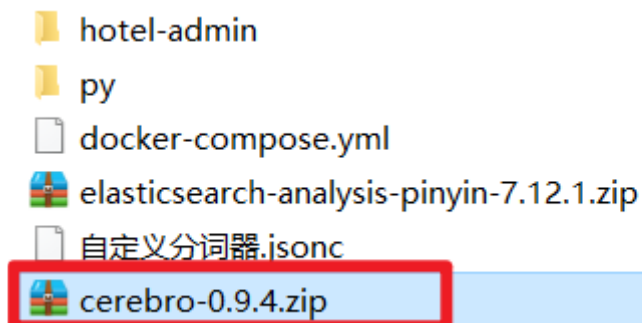
```
1 docker-compose up -d
```

4.2. 集群状态监控

kibana可以监控es集群，不过新版本需要依赖es的x-pack 功能，配置比较复杂。

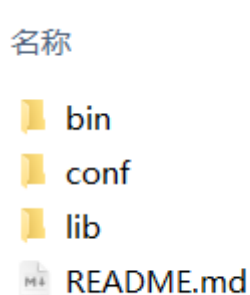
这里推荐使用cerebro来监控es集群状态，官方网址：<https://github.com/lmenezes/cerebro>

课前资料已经提供了安装包：

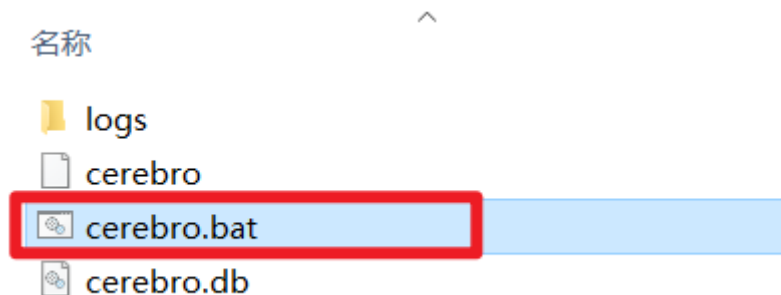


解压即可使用，非常方便。

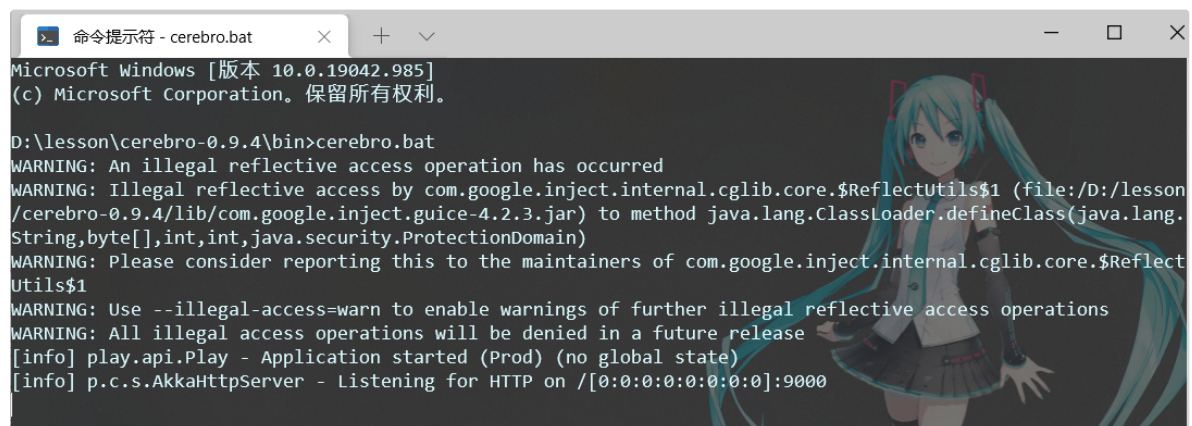
解压好的目录如下：



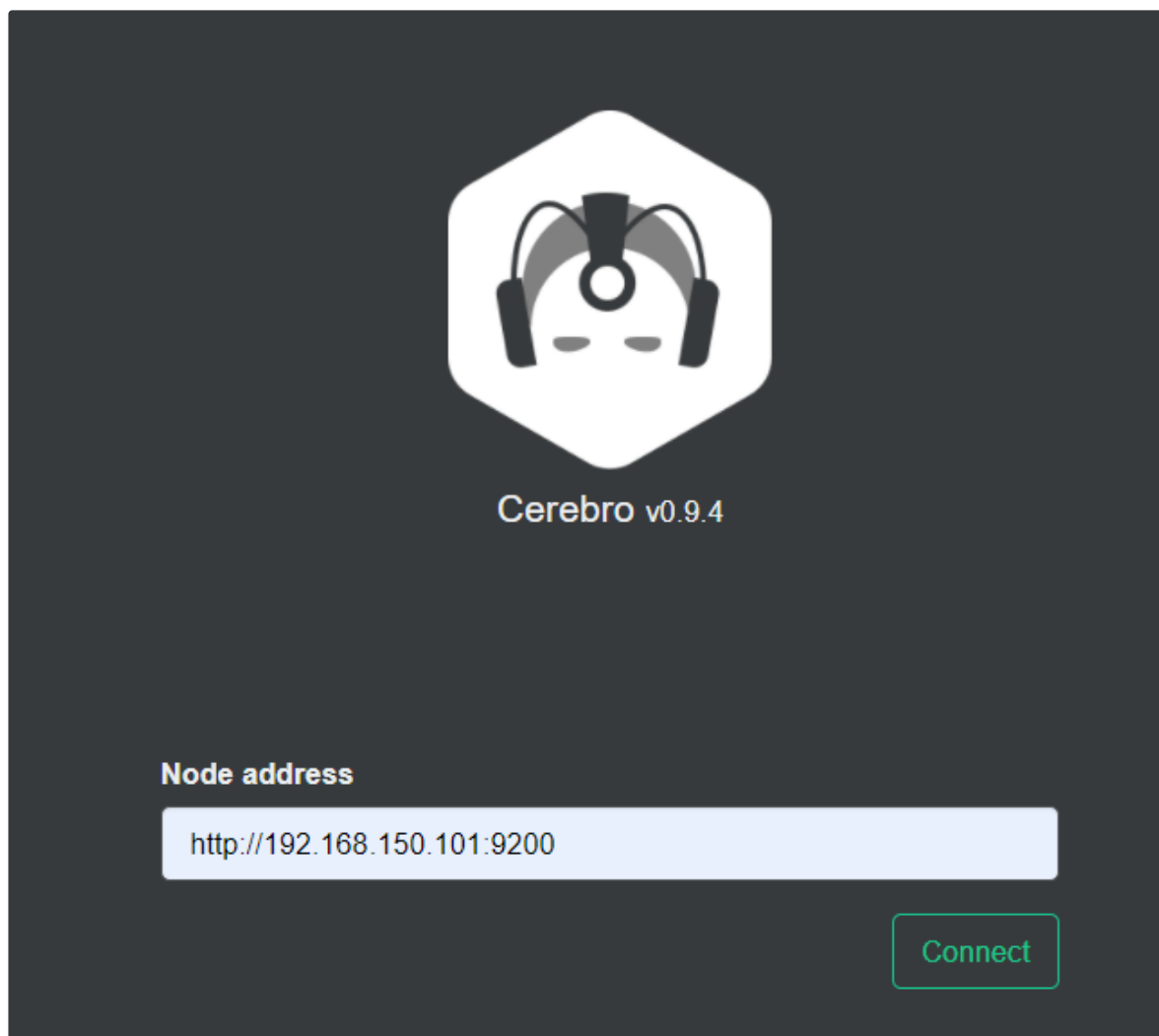
进入对应的bin目录：



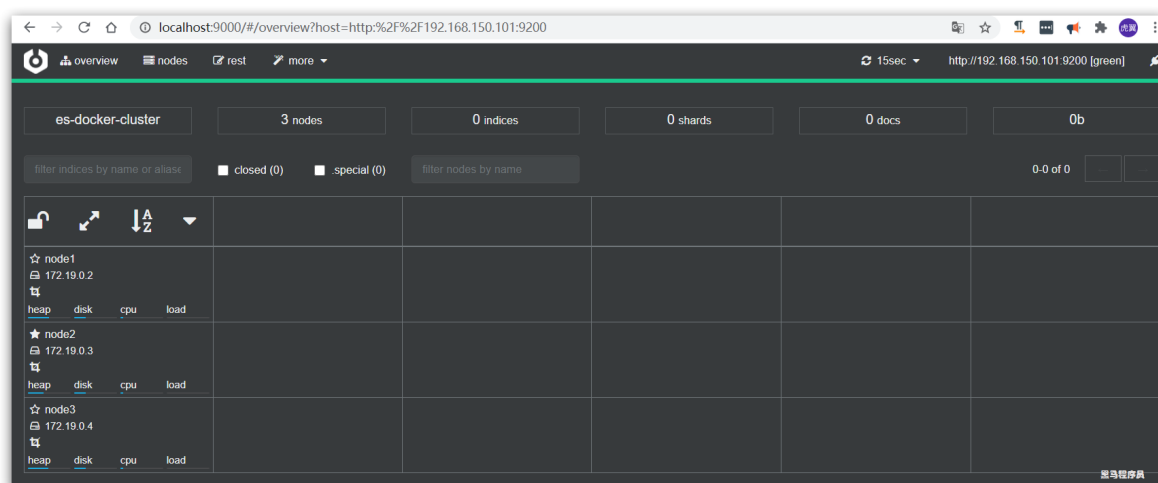
双击其中的cerebro.bat文件即可启动服务。



访问 <http://localhost:9000> 即可进入管理界面：



输入你的elasticsearch的任意节点的地址和端口，点击connect即可：



绿色的条，代表集群处于绿色（健康状态）。

4.3. 创建索引库

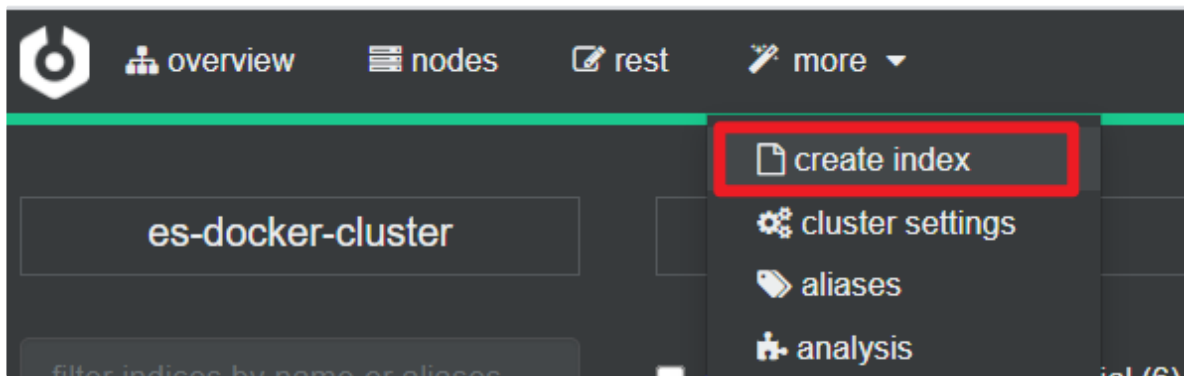
1) 利用kibana的DevTools创建索引库

在DevTools中输入指令：

```
1 PUT /itcast
2 {
3   "settings": {
4     "number_of_shards": 3, // 分片数量
5     "number_of_replicas": 1 // 副本数量
6   },
7   "mappings": {
8     "properties": {
9       // mapping映射定义 ...
10    }
11  }
12 }
```

2) 利用cerebro创建索引库

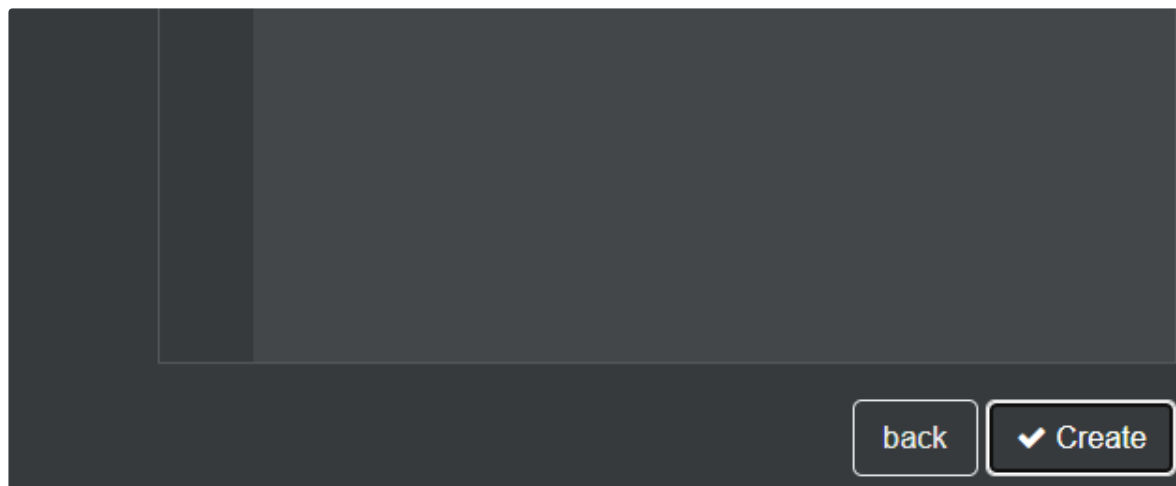
利用cerebro还可以创建索引库:



填写索引库信息:

A screenshot of the Cerebro web interface showing the 'create index' form. The form has the following fields: 'name' with the value 'itcast', 'number of shards' with the value '3', 'number of replicas' with the value '1', and a dropdown menu labeled 'load settings from existing index' which is currently empty.

点击右下角的create按钮：



4.4. 查看分片效果

回到首页，即可查看索引库分片效果：

