**ElasticSearch简单教程**

# 1简介

## 1.1 简介

ElasticSearch是一个基于Lucene的搜索服务器。它提供了一个分布式多用户能力的全文搜索引擎，基于RESTful web接口。Elasticsearch是用Java开发的，并作为Apache许可条款下的开放源码发布，是当前流行的企业级搜索引擎。设计用于云计算中，能够达到实时搜索，稳定，可靠，快速，安装使用方便。

解释上面一段话

上面的那一段话是从百度百科上摘下来的，相信大家和我一样，看了这样一段话大家都很懵。没关系，我们来一句一句解释它们。

lucene:通俗的讲，这是一个用来搜索的引擎，可以简单的理解为，一个数据库，只用来存文档的，但是存不是它的特点，快速的查找存进去的东西是他的特点。而为了快速的查找里面的东西，他会建立一个倒排索引 。

基于Restful接口：这个是一种框架风格，可以简单的理解为，用http请求的URL可以操作这个索引数据库。

## 1.2 使用场景

* 全文搜索
* 多数据源
* 读数据库

## 1.3 原理学习

参考：developer.51cto.com/art/201904/594615.htm

## 1.4 倒排索引

倒排索引（Inverted Index）也叫反向索引，有反向索引必有正向索引。通俗地来讲，正向索引是通过key找value，反向索引则是通过value找key。

参考：<https://www.cnblogs.com/cjsblog/p/10327673.html>

<https://blog.csdn.net/csdnliuxin123524/article/details/91581209>

# 2 安装

## 2.1 Window下安装步骤

因为是基于java开发的，那么安装elastic search之前jdk一定是要装好的。

接下来直接从官网上根据操作系统以及版本下载所需要的安装包。

下载地址：https://www.elastic.co/cn/downloads

下载并解压以后就可以使用了，没有听错，就是可以直接使用了，没有听错，开箱即用就是elasticsearch的特点之一。

在解压的目录下找到elasticsearch.bat文件双击即可，在浏览器中输入URL地址

http://locathost:9200

如果出现以一个json格式的数据表示安装成功了。



## 2.2 docker 安装

参考：<https://www.elastic.co/guide/en/elasticsearch/reference/7.4/docker.html>

下载镜像：docker pull docker.elastic.co/elasticsearch/elasticsearch:7.4.1

启动容器运行：docker run -p 9200:9200 -p 9300:9300 -e "discovery.type=single-node" docker.elastic.co/elasticsearch/elasticsearch:7.4.1

## 2.3 centos7安装

### 2.3.1 创建用户并设置密码

注意：root用户不能启动ES，所以需要创建新用户

使用root登录

创建用户esuser 命令如下：

adduser esuser

设置密码 命令如下：

passwd esuser

我的密码：Guo!123456

### **2.3.2 下载文件**

wget https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-7.4.0-linux-x86\_64.tar.gz

或者使用esuser 用户登录xftp，上传文件

### **2.3.3 使用xshell登录虚拟机，并切换到esuser用户 ，命令如下**

su esuser

### **2.3.4 解压文件，命令如下：**

tar xvzf elasticsearch-7.4.0-linux-x86\_64.tar.gz

//1.6 授权chown -R esuser elasticsearch-7.3.1

### **2.3.5 启动elasticsearch 进入bin目录，命令如下**

./elasticsearch

### **2.3.6 查看是否成功访问**

[http://localhost:9200/](http://localhost:9200/_cat/nodes?v)

查看当前节点信息

<http://localhost:9200/_cat/nodes?v>

# 3 centos7启动错误解决

问题出现环境，OS版本：CentOS-7-x86\_64-Minimal-1708；ES版本：elasticsearch-6.2.2。

## 3.1、max file descriptors [4096] for elasticsearch process is too low, increase to at least [65536]

每个进程最大同时打开文件数太小，可通过下面2个命令查看当前数量

修改/etc/security/limits.conf文件，增加配置，用户退出后重新登录生效

\* soft nofile 65536

\* hard nofile 65536

## 3.2 max virtual memory areas vm.max\_map\_count [65530] is too low, increase to at least [262144]

**修改/etc/sysctl.conf文件，增加配置vm.max\_map\_count=262144**

**vim /etc/sysctl.conf**

**sysctl -p**

**执行命令sysctl -p生效**

## 3.3 启动报错信息如下：

**解决：**

**vim /etc/elasticsearch/elasticsearch.yml**

**# 修改【#cluster.initial\_master\_nodes: ["node-1", "node-2"] 】**

**cluster.initial\_master\_nodes: ["node-1"]**

## 3.5 正常后访问

curl "http://127.0.0.1:9200" 能够正常访问，可是使用外网ip就提示拒绝链接

解决办法：vim config/elasticsearch.yml

增加：network.host: 0.0.0.0

# 4 Elasticsearch 目录结构如下

* bin ：脚本文件，包括 ES 启动 & 安装插件等等
* config ： elasticsearch.yml（ES 配置文件）、jvm.options（JVM 配置文件）、日志配置文件等等
* JDK ： 内置的 JDK，JAVA\_VERSION="12.0.1"
* lib ： 类库
* logs ： 日志文件
* modules ： ES 所有模块，包括 X-pack 等
* plugins ： ES 已经安装的插件。默认没有插件
* data ： ES 启动的时候，会有该目录，用来存储文档数据。该目录可以设置

# 5 单机集群多个 ES 实例安装

单机多个 ES 实例，形成一个 ES 单机伪集群，启动脚本如下：

|  |
| --- |
| bin/elasticsearch -E node.name=node01 -E cluster.name=bysocket\_es\_cluster -E path.data=node01\_data -d  bin/elasticsearch -E node.name=node02 -E cluster.name=bysocket\_es\_cluster -E path.data=node02\_data -d  bin/elasticsearch -E node.name=node03 -E cluster.name=bysocket\_es\_cluster -E path.data=node03\_data -d  bin/elasticsearch -E node.name=node04 -E cluster.name=bysocket\_es\_cluster -E path.data=node04\_data -d |

命令简单解释如下：

|  |
| --- |
| node.name ： ES 节点名称，即实例名  cluster.name ： ES 集群名称  path.data ： 指定了存储文档数据目录 |

执行完脚本后，需要等一会 ES 启动，也可以查看 logs 看看执行情况。

如何关闭集群中的 ES 实例，可以使用简单的命令实现：

|  |
| --- |
| ps | grep elasticsearch  kill -9 pid |

# 6 插件概述

## 6.1 简介

插件是用来增强 Elasticsearch 功能的方法，分为 核心插件（官方） & 社区插件。

## 6.2 安装及删除

安装 analysis-icu ICU 分析插件，命令如下：

|  |
| --- |
| sudo bin/elasticsearch-plugin install analysis-icu  本地安装示例如下：  bin/elasticsearch-plugin install file:///root/software/elasticsearch/analysis-smartcn-7.4.0.zip |

查看已安装的插件，命令如下：

|  |
| --- |
| bin/elasticsearch-plugin list  或者curl <http://localhost:9200/_cat/plugins> |

删除已安装的插件，命令如下：

|  |
| --- |
| sudo bin/elasticsearch-plugin remove analysis-icu |

## 6.3 analysis-icu 插件介绍

<https://blog.csdn.net/weixin_42257250/article/details/97757295>

## 6.4 分词器简介：

<https://blog.csdn.net/moshowgame/article/details/99448661>

## 6.5 简体中文插件

Smart Chinese Analyzer Plugins

中文分词器，听说Elastic Stack 8.0会自带，但是还没release，静候佳音吧。

Smart Chinese Analysis插件将Lucene的Smart Chinese分析模块集成到elasticsearch中。

提供中文或混合中英文本的分析器。 该分析器使用概率知识来查找简体中文文本的最佳分词。 首先将文本分成句子，然后将每个句子分割成单词。

sudo bin/elasticsearch-plugin install analysis-smartcn

本地安装：

./elasticsearch-plugin install [file:///root/software/elasticsearch/analysis-smartcn-7.4.0.zip](file://root\\software\\elasticsearch\\analysis-smartcn-7.4.0.zip)

查看分词情况：

|  |
| --- |
| curl -X POST "localhost:9200/\_analyze?pretty" -H 'Content-Type: application/json' -d'  {  "tokenizer": "standard",  "text": "中国向日葵"  }  ' |

|  |
| --- |
| curl -X POST "localhost:9200/\_analyze?pretty" -H 'Content-Type: application/json' -d'  {  "tokenizer": "smartcn\_tokenizer",  "text": "中国向日葵"  }  ' |

|  |
| --- |
| curl -X POST "localhost:9200/\_analyze?pretty" -H 'Content-Type: application/json' -d'  {  "tokenizer": "icu\_tokenizer",  "text": "中国向日葵"  }  ' |

# 7 RestApi介绍

参考：<https://www.elastic.co/guide/en/elasticsearch/reference/7.4/rest-apis.html>

# 8 java客户端

## 8.1 TransportClient 客户端

官方已经明确表示在ES 7.0版本中将弃用TransportClient客户端，且在8.0版本中完全移除它，所以不学习这个客户端。

## 8.2 Java REST Client

官方也给出了基于HTTP的客户端REST Client(推荐使用)，官方给出来的REST Client有Java Low Level REST Client和Java Hight Level REST Client两个，前者兼容所有版本的ES，后者是基于前者开发出来的，只暴露了部分API。

java rest client有两个实现类，

分别是RestClient和 RestHighLevelClient。

前者是一个低级客户端，通过Http与elasticsearch集群进行通信，可以做到 负载均衡、故障转移、持久化链接、自动发现集群节点等功能，同时支持所有elasticsearch版本，但是需要自己对请求和相应做编解码（自己写JSON）； 后者是一个高级客户端，对增删改差进行了封装，不需要处理编解码，类似之前的TransportClient，但是兼容性较差，对客户端和集群版本要求较高。  
 因为RestClient没有提供增删改差方法，只能自己写json并选择Http请求的方法进行实现，一般使用较少，只有RestHighLevelClient无法满足的情况下才会使用。这里主要介绍RestHighLevelClient。

8.3 操作步骤

引入相应依赖

<dependency>  
 <groupId>org.elasticsearch.client</groupId>  
 <artifactId>elasticsearch-rest-high-level-client</artifactId>  
 <version>7.4.0</version>

<!-- 注意排除依赖-->  
 <exclusions>  
 <exclusion>  
 <groupId>org.elasticsearch</groupId>  
 <artifactId>elasticsearch</artifactId>  
 </exclusion>  
 </exclusions>  
</dependency>  
<dependency>  
 <groupId>org.elasticsearch</groupId>  
 <artifactId>elasticsearch</artifactId>  
 <version>7.4.0</version>  
</dependency>

代码中声明客户端

public class RestHighLevelClientUtil {  
 private static final String *HOST\_NAME*="localhost";  
 private static final int *PORT*=9200;  
 private static final String *SCHEME*="http";  
 //获取客户端对象  
 public static RestHighLevelClient getClient(){  
 return new RestHighLevelClient(  
 RestClient.*builder*(  
 new HttpHost(*HOST\_NAME*, *PORT*, *SCHEME*)));  
  
 }  
}

然后就可进行创建索引，删除索引等各种操作了，具体见代码

8.3 官方文档

<https://www.elastic.co/guide/en/elasticsearch/client/java-rest/7.4/java-rest-high.html>

查看官方文档很重要\*\*\*\*\*\*\*\*\*\*\*\*\*\*