ELK 学习

1. ES 分词插件

Elasticsearch 中，内置了很多分词器（analyzers），例如 standard （标准分词器）、english （英文分词）和 chinese （中文分词）。其中 standard 就是无脑的一个一个词（汉字）切分，所以适用范围广，但是精准度低；english 对英文更加智能，可以识别单数负数，大小写，过滤停用词（例如“the”这个词）等；chinese 虽然是针对中文的分词器，但是效果很差，因此一般有中文分词需求时都会安装第三方分词插件，例如 ik、jieba、ansj 这些。

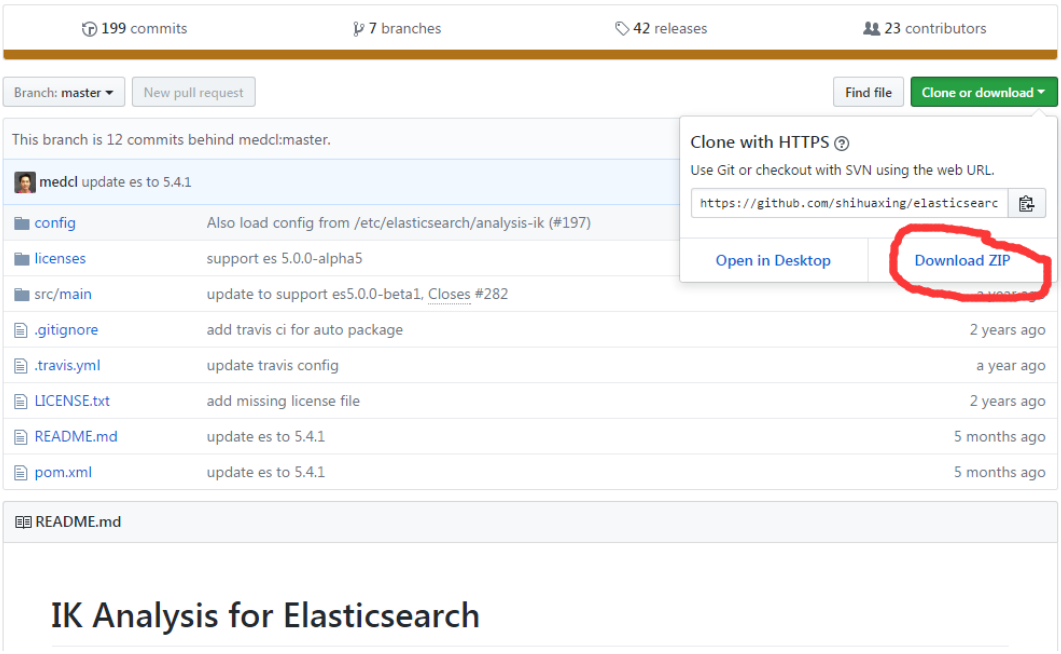
假设现在我们要索引进 ES 的文档中包含 我是中国人 这句话，以 ES 自带的中文分词器为例，通过以下请求：
http://localhost:9200/index/_analyze?analyzer=standard&pretty=true&text=我是中国人
我们会得到这样的结果

```
{
  tokens: [
    {
      token: textstart_offset: 2end_offset: 6type: <ALPHANUM>position: 1
    }{
      token: 我start_offset: 9end_offset: 10type: <IDEOGRAPHIC>position: 2
    }{
      token: 是start_offset: 10end_offset: 11type: <IDEOGRAPHIC>position: 3
    }{
      token: 中start_offset: 11end_offset: 12type: <IDEOGRAPHIC>position: 4
    }{
      token: 国start_offset: 12end_offset: 13type: <IDEOGRAPHIC>position: 5
    }{
      token: 人start_offset: 13end_offset: 14type: <IDEOGRAPHIC>position: 6
    }
  ]
}
```

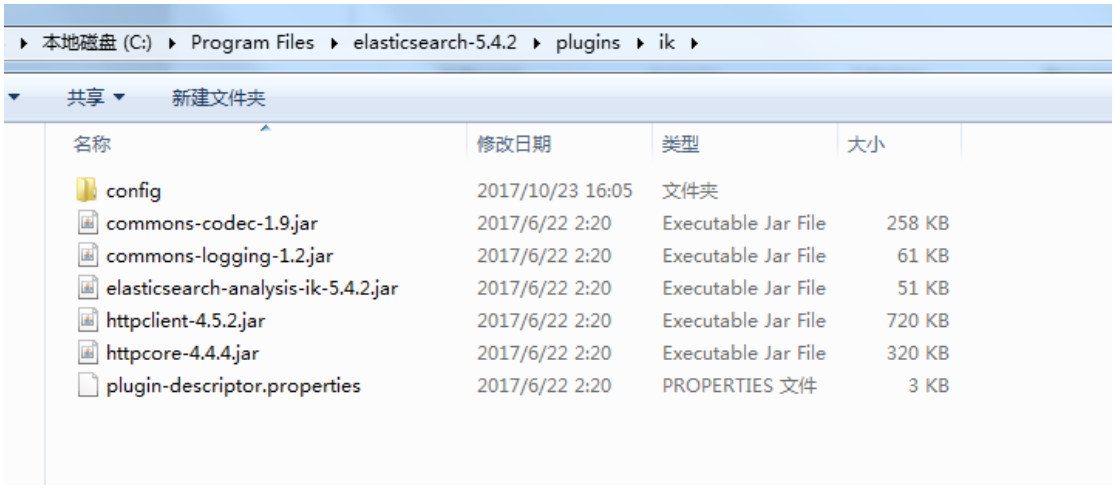
正常情况下，这不是我们想要的结果，比如我们更希望得到 “中国人”，“中国”，“我”这样的分词，这样我们就需要安装额外的中文分词插件。

插件的安装步骤很简单，直接从 GitHub 上下载压缩包后（也可以自己通过源码编译得到），解压到 your-es-root/plugins/目录下，这个目录专门用来存放 ES 相关的插件。

以 IK 分词插件为例，我们打开 github 主页，直接下载压缩包（Linux 可通过 wget 指令）：



然后将压缩包解压到 plugins 目录下



最后我们只要重启 ES 即可生效。接下来验证一下插件的效果：

（1）创建一个新的索引：

```
curl -XPUT http://localhost:9200/index
```

（2）在刚才的索引中创建一个映射规则，红色部分即指定该字段使用的分词器，其中 IK 定义两种分词模式：

- ik_max_word：会将文本做最细粒度的拆分，例如「中华人民共和国国歌」会被拆分为「中华人民共和国、中华人民、中华、华人、人民共和国、人民、人、民、共和国、共和、和、国国、国歌」，会穷尽各种可能的组合；

- ik_smart：会将文本做最粗粒度的拆分，例如「中华人民共和国国歌」会被拆分为「中华人民共和国、国歌」；

```
curl -XPOST http://localhost:9200/index/fulltext/_mapping -d'
   {
           "properties": {
```

```
            "content": {
                "type": "text",
                "analyzer": "ik_max_word",
                "search_analyzer": "ik_max_word"
            }
        }

}'
```

（3）对刚才的语句进行分词结果展示

```
curl -XPOST
"http://localhost:9200/index/_analyze?analyzer=ik_max_word&pretty=tru
e&text=我是中国人"
```

```
{
  tokens: [
    {
      token: textstart_offset: 2end_offset: 6type: ENGLISHposition: 1
    }{
      token: 我start_offset: 9end_offset: 10type: CN_CHARposition: 2
    }{
      token: 中国人start_offset: 11end_offset: 14type: CN_WORDposition: 3
    }{
      token: 中国start_offset: 11end_offset: 13type: CN_WORDposition: 4
    }{
      token: 国人start_offset: 12end_offset: 14type: CN_WORDposition: 5
    }
  ]
}
```

可以看到，分词结果比较符合语义，并且还去除了停用词“是”。

因为在中文分词中词典很重要，可以定制一些与具体语言场景相关的词组，在 IK 中我们可以通过修改 IKAnalyzer.cfg.xml 配置文件进行词典配置，IKAnalyzer.cfg.xml 一般存放在{conf}/analysis-ik/config/IKAnalyzer.cfg.xml o

或者{plugins}/elasticsearch-analysis-ik-*/config/IKAnalyzer.cfg.xml 目录下

simple_cc.data | psl.properties | IKAnalyzer.cfg.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
    <comment>IK Analyzer 扩展配置</comment>
    <!--用户可以在这里配置自己的扩展字典 -->
    <entry key="ext_dict">custom/mydict.dic;custom/single_word_low_freq.dic</entry>
     <!--用户可以在这里配置自己的扩展停止词字典-->
    <entry key="ext_stopwords">custom/ext_stopword.dic</entry>
    <!--用户可以在这里配置远程扩展字典 -->
    <!-- <entry key="remote_ext_dict">words_location</entry> -->
    <!--用户可以在这里配置远程扩展停止词字典-->
    <!-- <entry key="remote_ext_stopwords">words_location</entry> -->
</properties>
```

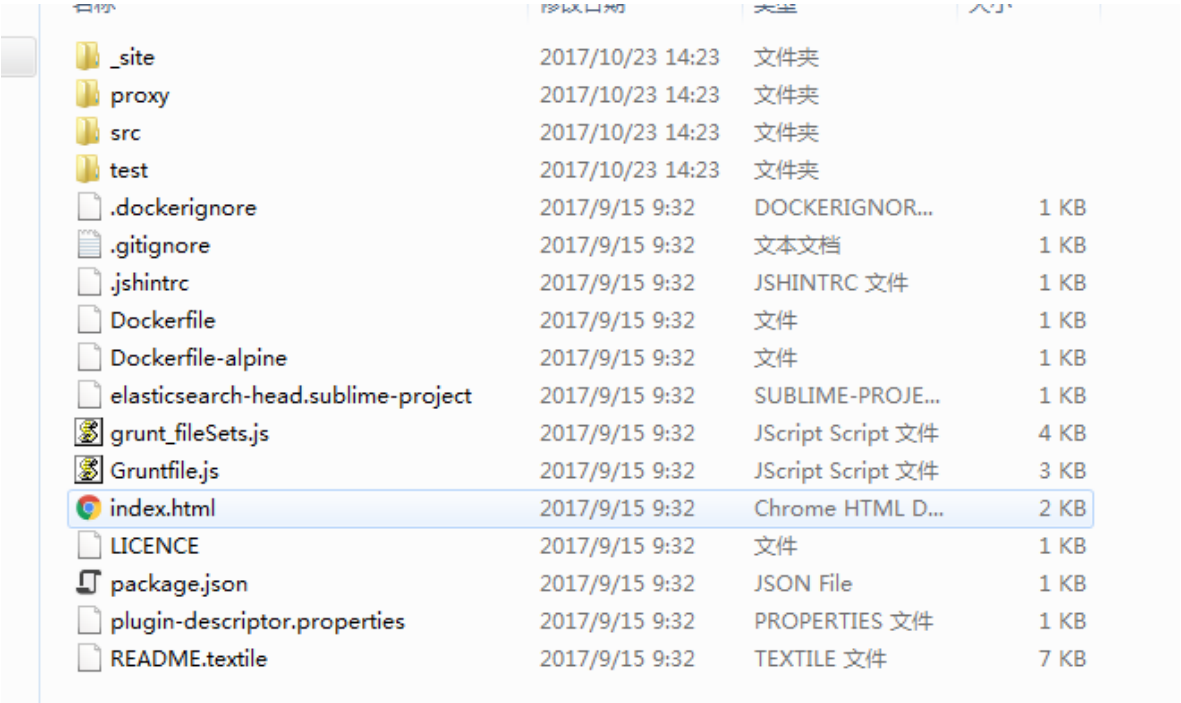
其它分词插件的安装使用过程基本与 IK 相同。

2. ES 可视化插件 elasticsearch-head

在学习 Elasticsearch 的过程中，必不可少需要通过一些工具查看 es 的运行状态以及数据。如果都是通过 rest 请求，未免太过麻烦，而且也不够人性化。此时，head 可以完美的帮助你快速学习和使用 es。

Head 插件采用纯 H5 编写，是一个独立的 html 项目，由于 head 是通过调用 rest 接口去获取并操作 es 状态和信息的，因此可以单独部署在 web 服务器上，不需要以插件的形式进行安装。并且在 es5.x 版本以后也不再支持插件形式，建议通过单独的 web 服务器进行部署。

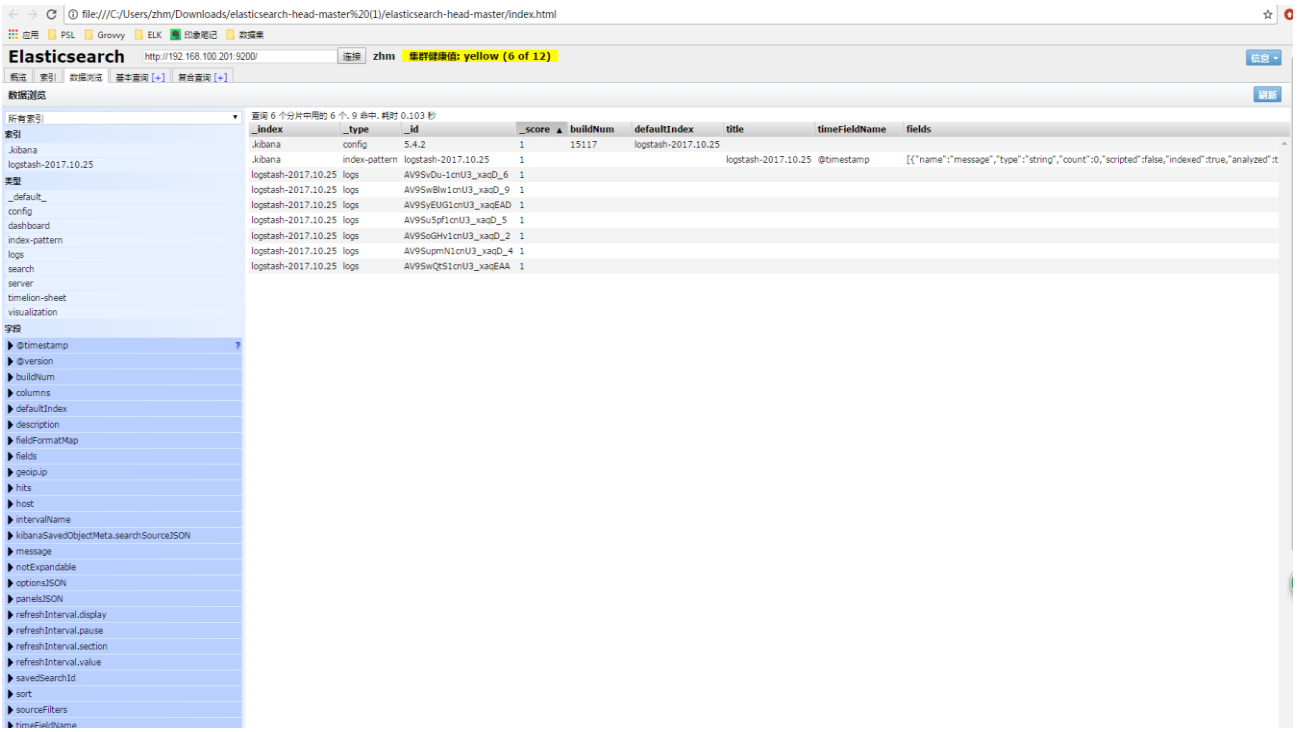
Head 插件可以在 GitHub 上下载到，解压后的内容如下



可以选择将其部署在常用的 Nginx、Tomcat 服务器上，通过 HTTP 远程访问，更方便的，我可以直接打开 index.html 文件，主页上输入我们要连接的 ES 地址，下面是在本机打开 index.html，连接服务器 http://192.168.100.201:9200/ 上启动的 ES，即可展示出集群的具体信息。

注：非插件的方式使用需要修改 ES 配置文件 elasticsearch.yml 中 cors 选项，在配置文件中加入

```
http.cors.enabled: true
http.cors.allow-origin: "*"
```



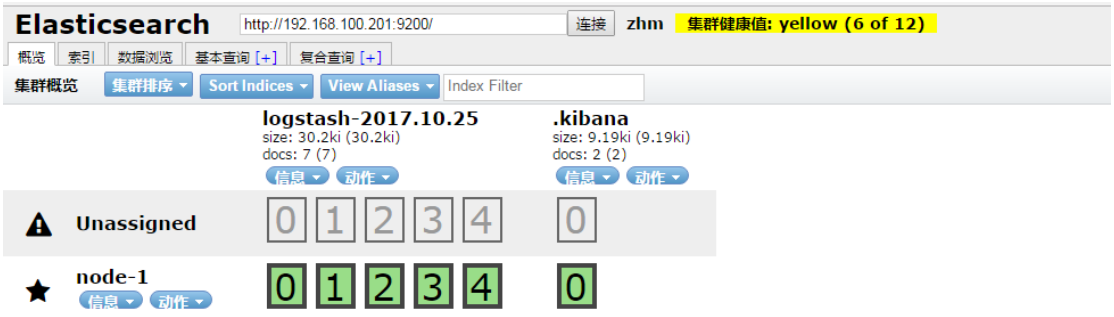
具体功能：

（1）概览

这个页面可以看到基本的分片的信息，比如主分片、副本分片等等，以及多少分片可用。

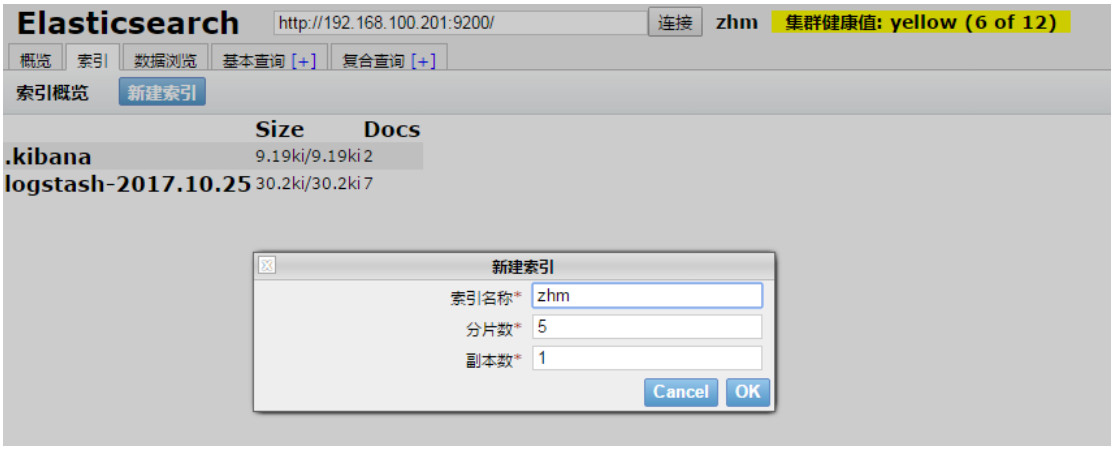
上方 zhm 是集群的名称，颜色表示集群的健康状态：

- 绿色表示主分片和副本分片都可用；
- 黄色表示只有主分片可用，没有副本分片；
- 红色表示主分片中的部分索引不可用，但是不耽误某些索引的访问。



（2）索引

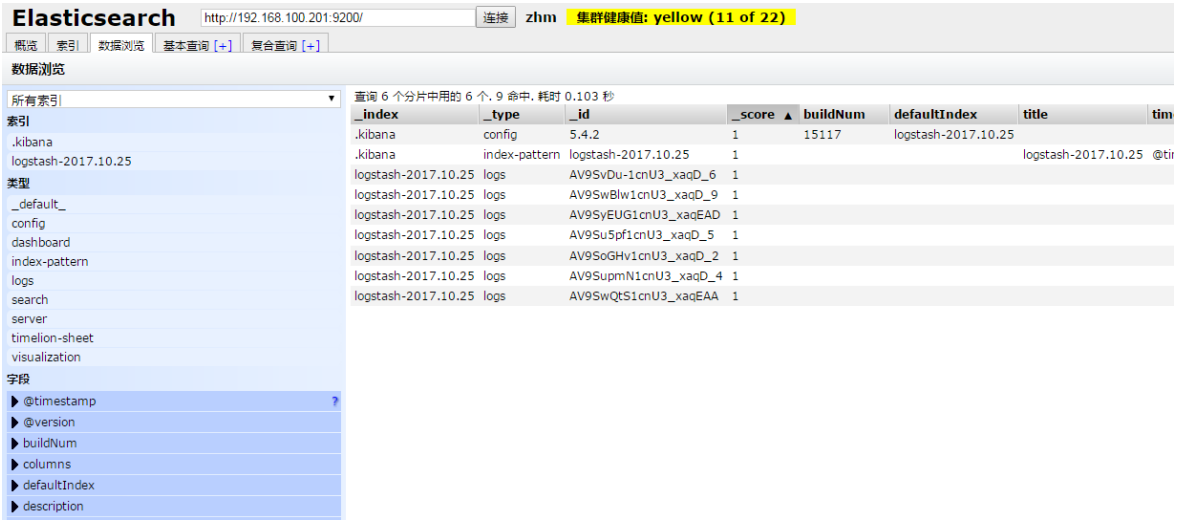
这个页面可以创建索引，并且可以设置分片的数量，副本的数量等等。



创建完索引，可以回来再看看概览的数据：

（3）数据浏览

这个 tab 页可以看到每个索引的基本信息，比如都有什么字段，存储的内容等等。不过这里并不能查询到全量的数据，想要看所有的数据，只能使用 scroll 进行分页查询。



（4）基本查询

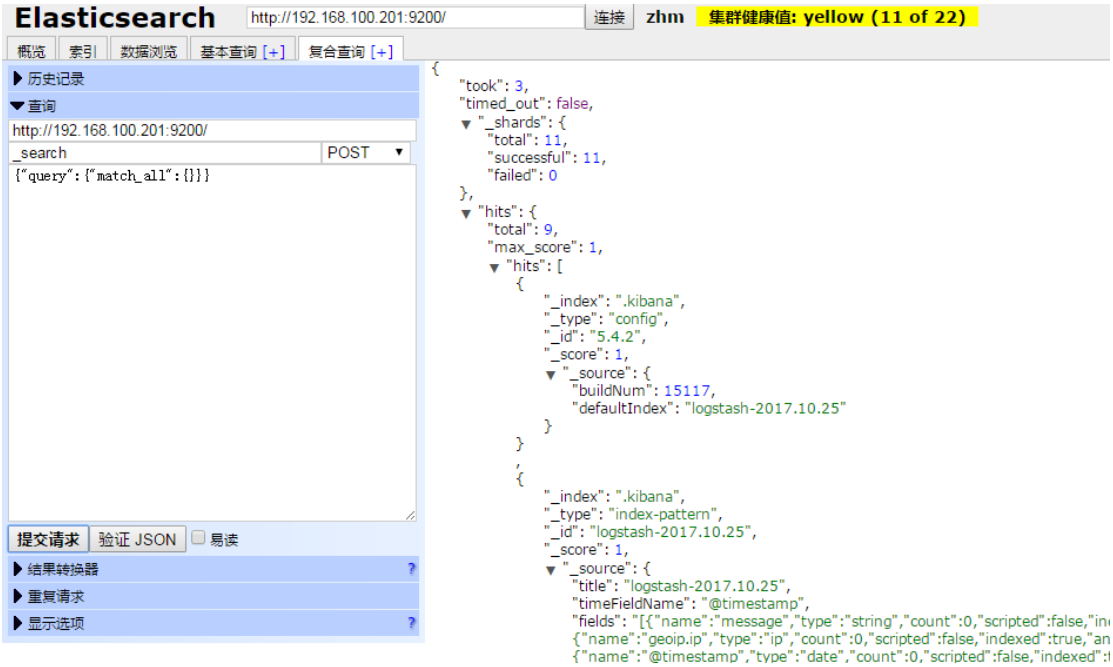
在这里可以拼接一些基本的查询。
如果了解 elasticsearch 的话，可以在这里轻松拼接出 es 的查询语句。

（4）复合查询

这里比较常用，不仅仅可以做查询，还可以执行 PUT DELETE 等 curl 的命令。

因此，刚学习 es 时，不需要在 windows 下安装 curl，直接在这里就可以提交一些 rest 请求。这里能使用的功能还是很多的，所有需要通过 curl 执行的 rest 请求，都可以在这里执行：

- 创建/删除索引
- 索引/更新/删除数据
- 创建映射
- 创建别名
- 指定路由 等等



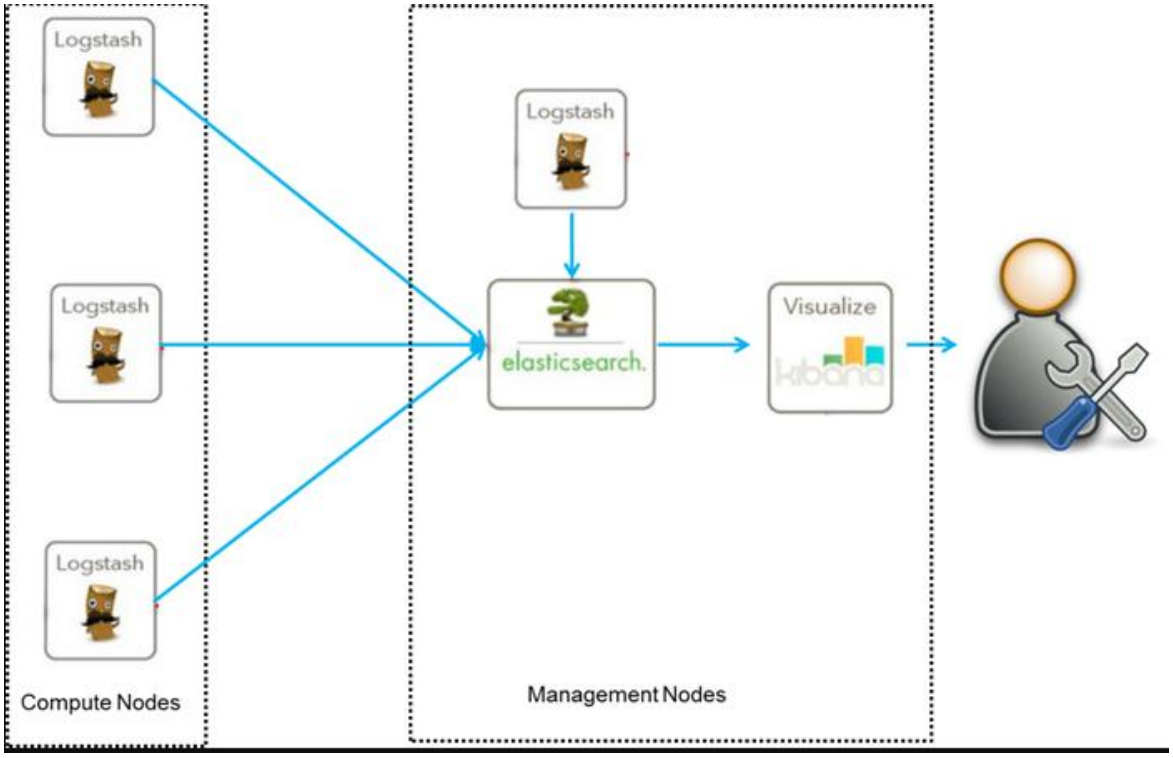
3. ELK 搭建（Elasticsearch、Logstash、Kibana）

ELK 是 Elasticsearch、Logstash、Kibana 的简称，这三者是核心套件，但并非全部。

Elasticsearch 是实时全文搜索和分析引擎，提供搜集、分析、存储数据三大功能；是一套开放 REST 和 JAVA API 等结构提供高效搜索功能，可扩展的分布式系统。它构建于 Apache Lucene 搜索引擎库之上。

Logstash 是一个用来搜集、分析、过滤日志的工具。它支持几乎任何类型的日志，包括系统日志、错误日志和自定义应用程序日志。它可以从许多来源接收日志，这些来源包括 syslog、消息传递（例如 RabbitMQ）和 JMX，它能够以多种方式输出数据，包括电子邮件、websockets 和 Elasticsearch。

Kibana 是一个基于 Web 的图形界面，用于搜索、分析和可视化存储在 Elasticsearch 指标中的日志数据。它利用 Elasticsearch 的 REST 接口来检索数据，不仅允许用户创建他们自己的数据的定制仪表板视图，还允许他们以特殊的方式查询和过滤数据



环境：201 服务器上单机版配置，ELK 版本均为 5.4.2

（1）安装 Elasticsearch

wget https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-5.4.2.zip

解压后进入 bin 目录运行./elasticsearch 即可。

修改配置文件 root-path/config/elasticsearch.yml 如下：

cluster.name: zhm

node.name: node-1

network.host: 127.0.0.1,192.168.100.201

（2）安装 Logstash

wget https://artifacts.elastic.co/downloads/logstash/logstash-5.4.2.zip

解压进入 bin 目录

简单地，可以通过以下命令启动 Logstash，这样的效果是标准输入 stdin 作为日志输入，标准输出 stduot 作为输出。

```
# ./logstash -e 'input { stdin { } } output { stdout {} }'
```

```
The stdin plugin is now waiting for input:
[2017-10-26T17:43:30,045][INFO ][logstash.agent           ] Successfully started Logsta
Hello,World!
2017-10-26T09:44:25.828Z deepintell-web-server Hello,World!
```

当需要从日志文件或其他方式读取数据后将数据转存到 ES 中，可以通过配置文件的形式进行定义：

注：

-e 执行操作

input 标准输入

{ stdin } 插件

output 标准输出

{ stdout } 插件

创建配置文件 elk.conf

```
# vim root-path/bin/elk.conf
```

文件中添加以下内容

```
input { stdin { } } output { elasticsearch { hosts => ["192.168.1.202:9200"] } stdout { codec => rubydebug } }
```

使用配置文件运行 logstash `# ./logstash -f ./elk.conf`（保证 ES 已经开启）

（3）安装 Kibana

wget https://artifacts.elastic.co/downloads/kibana/kibana-5.4.2-linux-x86_64.tar.gz

解压 `tar -xzf kibana-5.4.2-linux-x86_64.tar.gz`

进入 config 目录，编辑 kibana 的配置文件

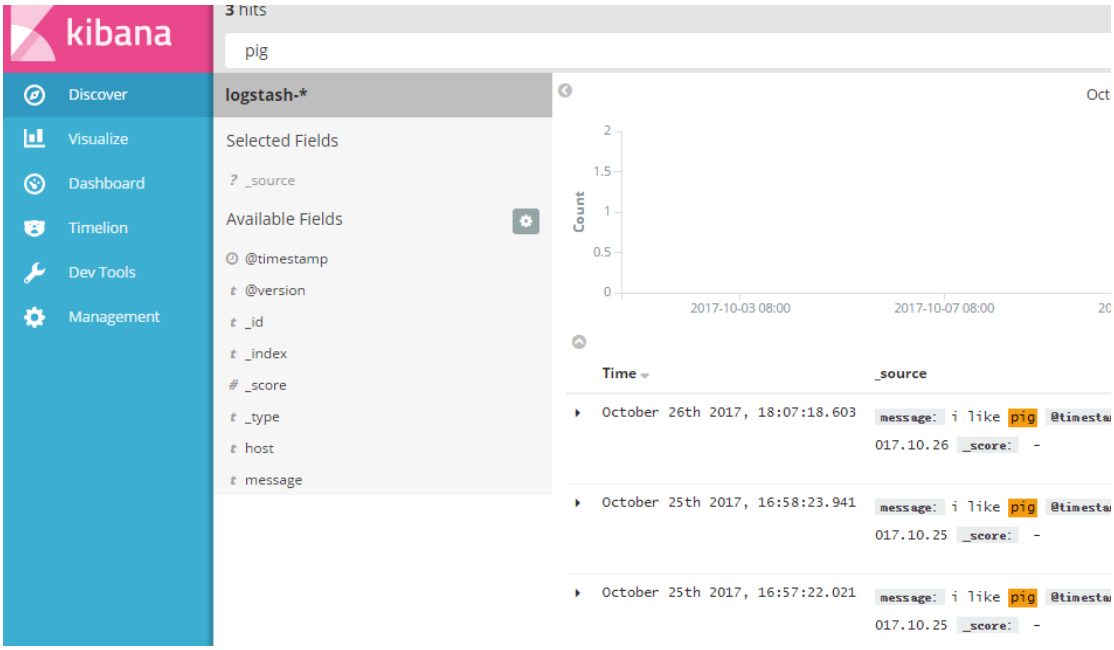
```
# vim  ./kibana.yml
```

修改配置文件如下，开启以下的配置

```
server.port: 5601  #kibana 服务器端口
server.host: "192.168.100.202" #kibana 服务器主机 ip
elasticsearch.url: "http://192.168.100.201:9200" #ES 服务地址
kibana.index: ".kibana" #kibana 在 ES 中需要建立的索引名称
```

安装完成后一次启动 Elasticsearch、Logstash、Kibana，由于 Logstash 配置为从标准输入读取数据，我们在这段输入消息并回车，消息就会经过 Logstash 发送到 ES 中，一般储存在 logstash-2017.10.25 这样格式的索引中

同时我们可以在 kibana 的界面上通过关键词搜索去查询我们关心的内容，



以上只是基本的搭建过程，ELK 中每个组件都有其详细的用途与功能。