# ElasticSearch介绍

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](https://baike.baidu.com/item/%E4%BA%91%E8%AE%A1%E7%AE%97" \t "http://blog.java1234.com/blog/articles/_blank)中，能够达到实时搜索，稳定，可靠，快速，安装使用方便。

我们建立一个网站或应用程序，并要添加搜索功能，但是想要完成搜索工作的创建是非常困难的。我们希望搜索解决方案要运行速度快，我们希望能有一个零配置和一个完全免费的搜索模式，我们希望能够简单地使用JSON通过HTTP来索引数据，我们希望我们的搜索服务器始终可用，我们希望能够从一台开始并扩展到数百台，我们要实时搜索，我们要简单的多租户，我们希望建立一个云的解决方案。因此我们利用Elasticsearch来解决所有这些问题以及可能出现的更多其它问题。

# Elasticsearch安装部署（centos7）

## 1必须要有jre支持

elasticsearch是用Java实现的，跑elasticsearch必须要有jre支持，所以必须先安装jre

在正式环境里 我们可以不安装jdk ，仅仅安装Java运行环境 jre即可；

第一步：下载jre

我们去oracle官方下载下jre

<http://www.oracle.com/technetwork/java/javase/downloads/server-jre8-downloads-2133154.html>

下载 server jre



第二步：上传jre到/home/data/下

rz命令

第三步：解压以及剪切到/home/java/目录

[root@localhost ~]# cd /home/data/

[root@localhost data]# ls

server-jre-8u131-linux-x64.tar.gz

[root@localhost data]# tar -zxvf server-jre-8u131-linux-x64.tar.gz

进入/home/data/ 解压tar.gz压缩包

[root@localhost data]# ls

jdk1.8.0\_131  server-jre-8u131-linux-x64.tar.gz

[root@localhost ~]# mkdir /home/java

[root@localhost ~]# ls /home/

data  java

[root@localhost ~]#

在/home下新建java文件夹，用来放jre

[root@localhost ~]# mv /home/data/jdk1.8.0\_131/ /home/java/

[root@localhost ~]# ls /home/java/

jdk1.8.0\_131

[root@localhost ~]#

把/home/data/下的jre剪切到/home/java/下

第四步：jre目录以及子目录授予root权限

chown root:root -R /home/java/jdk1.8.0\_131/

第五步：配置环境变量

编辑环境变量配置文件

vi /etc/profile

在文件末尾加入下面代码，强制保存（w!）退出

export JAVA\_HOME=/home/java

export JRE\_HOME=/home/java/jdk1.8.0\_131

export CLASSPATH=$JRE\_HOME/lib/rt.jar:$JRE\_HOME/lib/ext

export PATH=$PATH:$JRE\_HOME/bin

使环境变量即时生效

source /etc/profile

第六步：测试

[root@localhost ~]# java -version

java version "1.8.0\_131"

Java(TM) SE Runtime Environment (build 1.8.0\_131-b11)

Java HotSpot(TM) 64-Bit Server VM (build 25.131-b11, mixed mode)

以及java和javac命令都行

## 2下载elasticsearch

进入官方下载 <https://www.elastic.co/downloads/elasticsearch>

因为是centos中运行 所以我们选 tar.gz压缩包；

下载后 用ftp上传到centos里 我们把这个文件上传到 /home/data/下

## 3安装和配置elasticsearch

进行解压命令

tar -zxvf elasticsearch-6.2.4.tar.gz

启动命令（这里我是自己新建了目录）

sh /home/es/elasticsearch-6.2.4/bin/elasticsearch

sh /home/es/elasticsearch-6.2.4/bin/elasticsearch -d （后台启动）

## **4常见错误**

1. org.elasticsearch.bootstrap.StartupException: java.lang.RuntimeException: can not run elasticsearch as root

原因：意思是不能用root用户来启动

解决：那我们新建一个用户来启动

新增用户

useradd elastic

给新增的用户分配目录权限

chown -R elastic:elastic /home/es/elasticsearch-6.2.4/

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

解决：

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

\* soft nofile 65536

\* hard nofile 65536

1. max number of threads [3818] for user [es] is too low, increase to at least [4096]

解决：

问题同上，最大线程个数太低。修改配置文件/etc/security/limits.conf，增加配置

\*               soft    nproc           4096

\*               hard    nproc           4096

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

解决：

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

vi /etc/sysctl.conf

sysctl -p

　　 执行命令sysctl -p生效

## 安装部署成功效果

我们来验证下服务是否正常运行 curl [http://localhost:9200](http://localhost:9200/)

{

  "name" : "K22mJd5",

  "cluster\_name" : "elasticsearch",

  "cluster\_uuid" : "R2qfXKtrQl2PwKdJMmPuMA",

  "version" : {

    "number" : "5.5.2",

    "build\_hash" : "b2f0c09",

    "build\_date" : "2017-08-14T12:33:14.154Z",

    "build\_snapshot" : false,

    "lucene\_version" : "6.6.0"

  },

  "tagline" : "You Know, for Search"

}

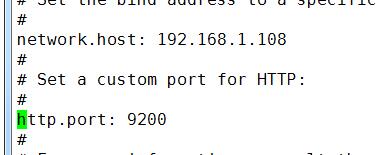
出来这个 说明配置OK。

## 允许外网连接配置

前面我们配置的仅仅是本机使用 但是我们比如集群以及其他机器连接 ，则需要配置下。

可以修改 /home/es/elasticsearch/config/elasticsearch.yml 文件

把 network.host 和 http.port 前面的 备注去掉 然后Host改成你的局域网IP即可



修改后 保存退出

再启动下elasticsearch

我们用谷歌浏览器请求下 <http://192.168.1.108:9200/>



OK 出现这东西 才算配置完成；

# Head插件安装

elasticsearch head插件是一个入门级的elasticsearch前端插件；我们来安装下

## 1.安装nodejs

head插件是nodejs实现的，所以必须先安装Nodejs

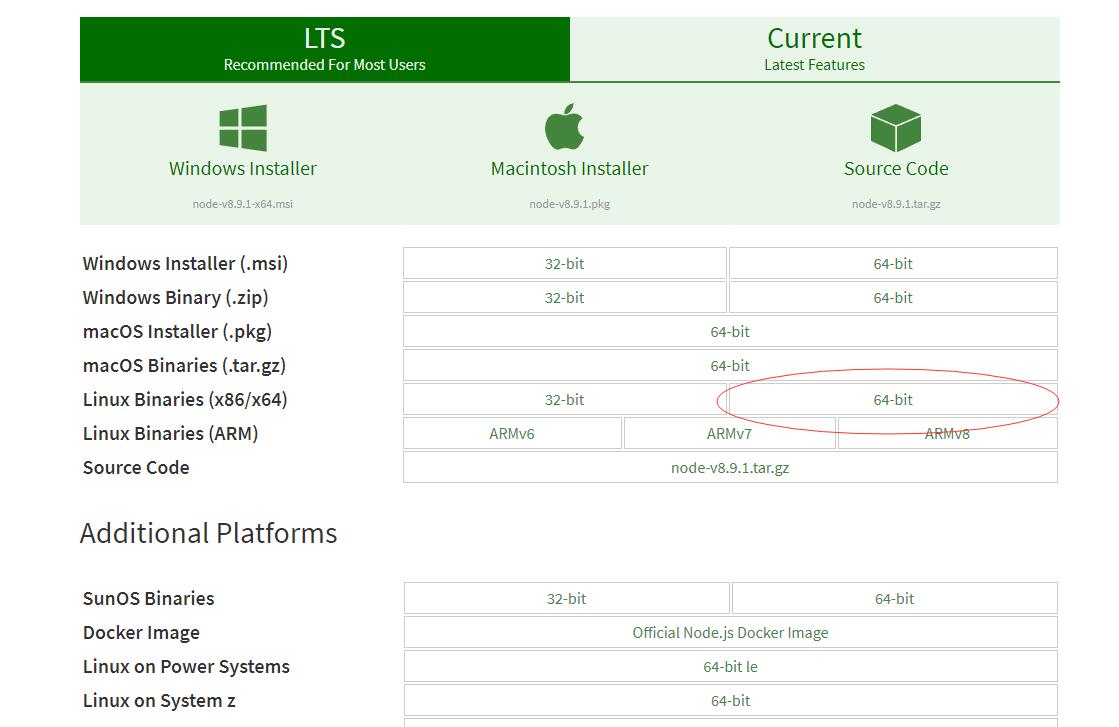
NodeJs安装有好几种方式：

第一种： 最简单的是用yum命令，可惜我现在用的时候 发现 镜像中没有nodejs；所以这种方式放弃；

第二种：去官网下载源码，然后自己编译；编译过程中可能会出现问题，嫌麻烦也放弃这种方式；

第三种：去官网直接下载编译好的二进制文件，这种最方便；我们使用第三种方式；

先去官网：[https://nodejs.org/en/download/](https://nodejs.org/en/download/" \o "https://nodejs.org/en/download/" \t "http://blog.java1234.com/blog/articles/_blank)



我们是centos 64位 所以右击 红圈里的 复制下载地址：

[https://nodejs.org/dist/v8.9.1/node-v8.9.1-linux-x64.tar.xz](https://nodejs.org/dist/v8.9.1/node-v8.9.1-linux-x64.tar.xz" \o "https://nodejs.org/dist/v8.9.1/node-v8.9.1-linux-x64.tar.xz" \t "http://blog.java1234.com/blog/articles/_blank)

我们准备把nodejs安装到usr/local 下

所以 先 cd /usr/local/ 进local目录

然后

wget

<https://nodejs.org/dist/v8.9.1/node-v8.9.1-linux-x64.tar.xz> 下载压缩包

xz文件 我们需要解压

tar -xJf node-v8.9.1-linux-x64.tar.xz

然后我们再配置环境变量：

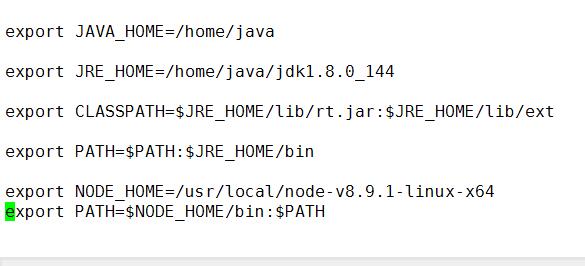
vi /etc/profile

打开

在最后加上：

export NODE\_HOME=/usr/local/node-v8.9.1-linux-x64

export PATH=$NODE\_HOME/bin:$PATH



加完之后，保存；

再执行 source /etc/profile  让环境变量生效；

这里再说下 老版本的nodejs是不自带的npm工具的（ NPM是随同NodeJS一起安装的包管理工具,能解决NodeJS代码部署上的很多问题） 以往都要单独下载安装，现在的话 新版本都自己npm 所以不需要再安装了；

我们来验证下：

[root@bogon node-v8.9.1-linux-x64]# node -v

v8.9.1

[root@bogon node-v8.9.1-linux-x64]# npm -v

5.5.1

说明安装OK；

## 2.安装git

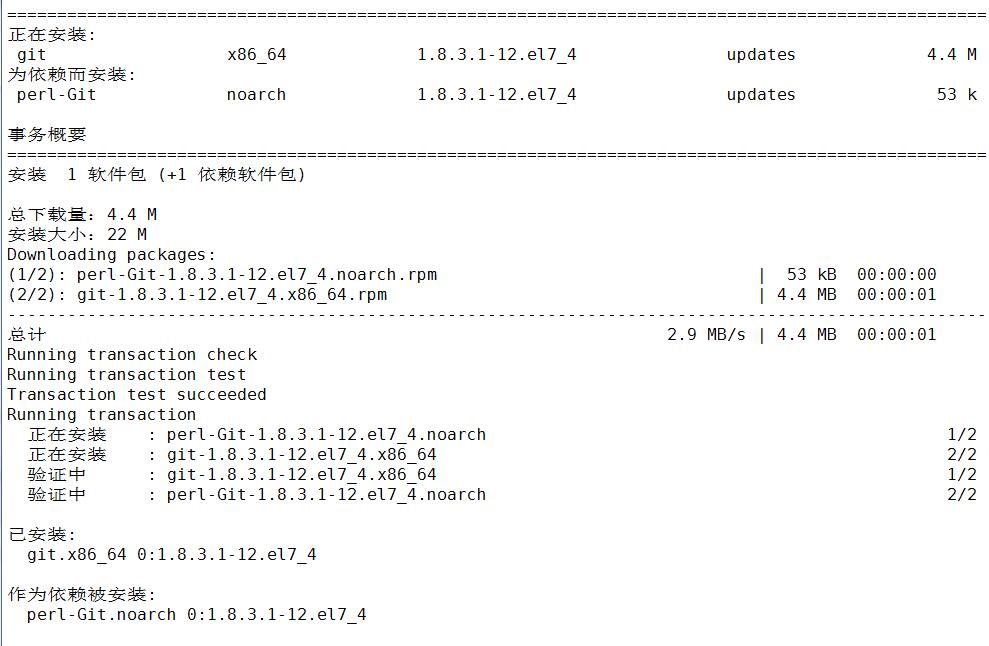
我们要用git方式下载head插件

git是一个分布式版本控制系统

我们在Centos上安装GIT有好几种方式。这里我们用最简单的yum命令方式 一步到位，省了下载编译安装这些繁琐过程。

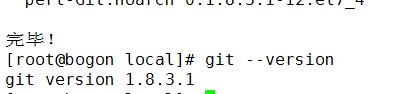
输入命令：

yum install -y git



验证：

git --version



## 3.下载以及安装head插件

打开 <https://github.com/mobz/elasticsearch-head>

Running with built in server

命令如下

git clone git://github.com/mobz/elasticsearch-head.git

cd elasticsearch-head

npm install

npm run start

open <http://localhost:9100/>

我们用这种最简单的方式安装

当然我们安装的地方是 /usr/local/

## 4.配置elasticsearch,允许head插件访问

进入elasticsearch安装目录下的config目录 打开 elasticsearch.yml

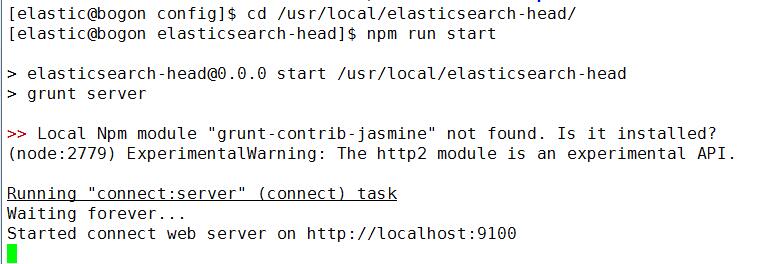
最后加上

http.cors.enabled: true

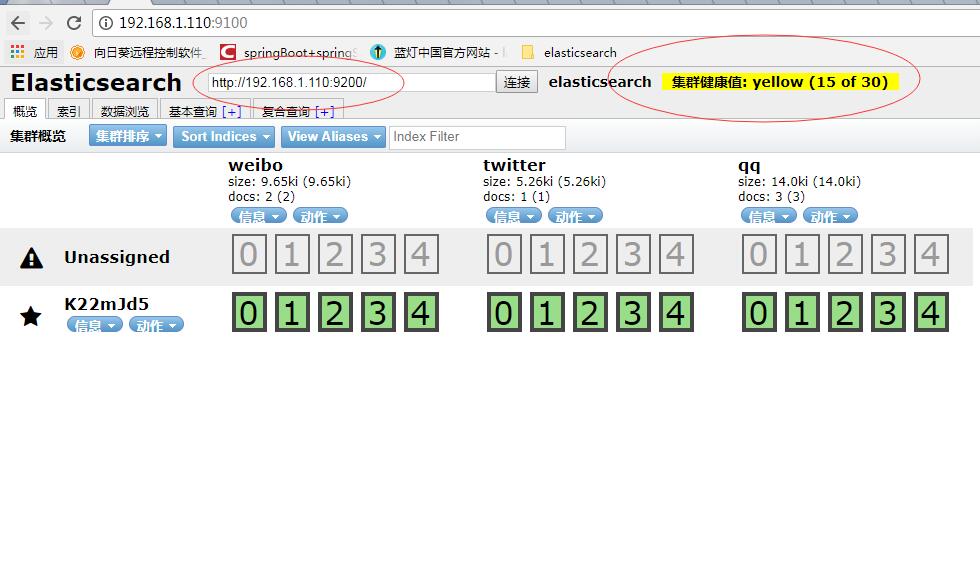
http.cors.allow-origin: "\*"

## 5.测试

启动elasticsearch，再进入head目录，执行npm run start 启动插件



说明启动成功，然后浏览器 执行 <http://192.168.1.110:9100/>



内部输入 <http://192.168.1.110:9200/> 点击连接 如果右侧输出黄色背景字体 说明配置完整OK；

# Elasticsearch集群配置

## elasticsearch多机集群配置

**配置集群之前  先把要加群集群的节点的里的data目录下的Node目录 删除，否则集群建立会失败。**

我这边虚拟机配置了两台centos IP分别是 192.168.1.110 和 192.168.1.111 ;

分别配置下elasticsearch.yml配置文件

110机器：

# ---------------------------------- Cluster -----------------------------------

#

# Use a descriptive name for your cluster:

#

**cluster.name: my-application**

#

# ------------------------------------ Node ------------------------------------

#

# Use a descriptive name for the node:

#

**node.name: node-1**

#

# Add custom attributes to the node:

#

#node.attr.rack: r1

#

# ----------------------------------- Paths ------------------------------------

#

# Path to directory where to store the data (separate multiple locations by comma):

#

#path.data: /path/to/data

#

# Path to log files:

#

#path.logs: /path/to/logs

#

# ----------------------------------- Memory -----------------------------------

#

# Lock the memory on startup:

#

#bootstrap.memory\_lock: true

#

# Make sure that the heap size is set to about half the memory available

# on the system and that the owner of the process is allowed to use this

# limit.

#

# Elasticsearch performs poorly when the system is swapping the memory.

#

# ---------------------------------- Network -----------------------------------

#

# Set the bind address to a specific IP (IPv4 or IPv6):

#

**network.host: 192.168.1.110**

#

# Set a custom port for HTTP:

#

**http.port: 9200**

#

# For more information, consult the network module documentation.

#

# --------------------------------- Discovery ----------------------------------

#

# Pass an initial list of hosts to perform discovery when new node is started:

# The default list of hosts is ["127.0.0.1", "[::1]"]

#

**discovery.zen.ping.unicast.hosts: ["192.168.1.110"]**

#

# Prevent the "split brain" by configuring the majority of nodes (total number of master-eligible nodes / 2 + 1):

#

#discovery.zen.minimum\_master\_nodes: 3

#

# For more information, consult the zen discovery module documentation.

#

# ---------------------------------- Gateway -----------------------------------

#

# Block initial recovery after a full cluster restart until N nodes are started:

#

#gateway.recover\_after\_nodes: 3

#

# For more information, consult the gateway module documentation.

#

# ---------------------------------- Various -----------------------------------

#

# Require explicit names when deleting indices:

#

#action.destructive\_requires\_name: true

http.cors.enabled: true

http.cors.allow-origin: "\*"

111机器：

# ---------------------------------- Cluster -----------------------------------

#

# Use a descriptive name for your cluster:

#

**cluster.name: my-application**

#

# ------------------------------------ Node ------------------------------------

#

# Use a descriptive name for the node:

#

**node.name: node-2**

#

# Add custom attributes to the node:

#

#node.attr.rack: r1

#

# ----------------------------------- Paths ------------------------------------

#

# Path to directory where to store the data (separate multiple locations by comma):

#

#path.data: /path/to/data

#

# Path to log files:

#

#path.logs: /path/to/logs

#

# ----------------------------------- Memory -----------------------------------

#

# Lock the memory on startup:

#

#bootstrap.memory\_lock: true

#

# Make sure that the heap size is set to about half the memory available

# on the system and that the owner of the process is allowed to use this

# limit.

#

# Elasticsearch performs poorly when the system is swapping the memory.

#

# ---------------------------------- Network -----------------------------------

#

# Set the bind address to a specific IP (IPv4 or IPv6):

#

**network.host: 192.168.1.111**

#

# Set a custom port for HTTP:

#

**http.port: 9200**

#

# For more information, consult the network module documentation.

#

# --------------------------------- Discovery ----------------------------------

#

# Pass an initial list of hosts to perform discovery when new node is started:

# The default list of hosts is ["127.0.0.1", "[::1]"]

#

**discovery.zen.ping.unicast.hosts: ["192.168.1.110"]**

#

# Prevent the "split brain" by configuring the majority of nodes (total number of master-eligible nodes / 2 + 1):

#

#discovery.zen.minimum\_master\_nodes: 3

#

# For more information, consult the zen discovery module documentation.

#

# ---------------------------------- Gateway -----------------------------------

#

# Block initial recovery after a full cluster restart until N nodes are started:

#

#gateway.recover\_after\_nodes: 3

这里两台机器的cluster.name必须一致 这样才算一个集群

node.name节点名称每台取不同的名称，用来表示不同的集群节点

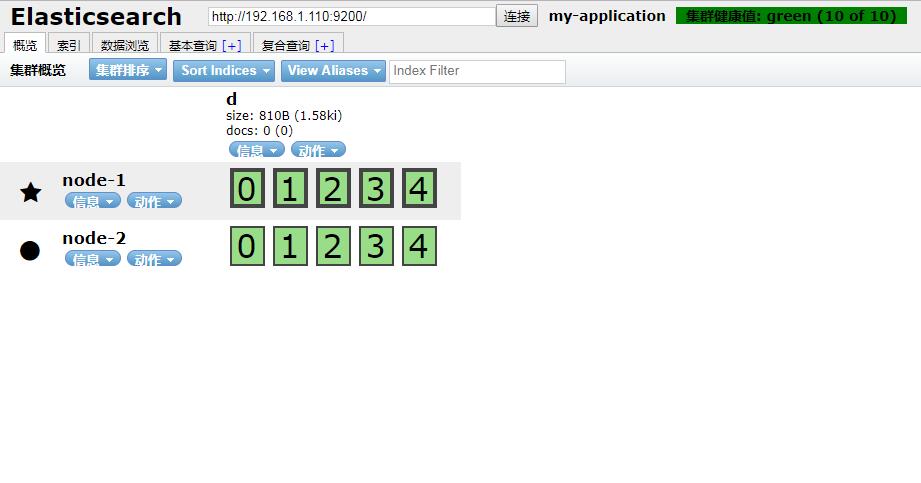
network.host配置成自己的局域网IP

http.port端口就固定9200

discovery.zen.ping.unicast.hosts主动发现节点我们都配置成110节点IP

配置完后 重启es服务；

然后head插件我们查看下：



说明集群配置OK 。

# 将数据库数据同步到elasticsearch

## 一、Logstash-6.2.4下载安装:

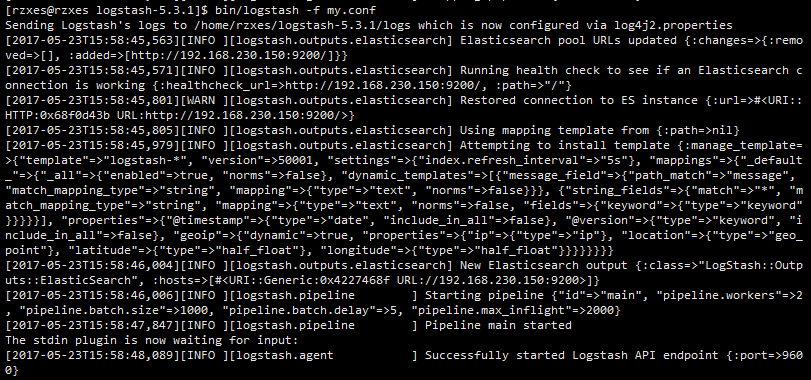
* 下载：<https://www.elastic.co/cn/downloads/logstash>
* 解压：tar -zxf logstash-6.2.4.tar.gz
* 启动：bin/logstash -e 'input { stdin { } } output { stdout {} }'  （参数表示终端输入输出）如下则成功。
* Sending Logstash's logs to /home/rzxes/logstash-6.2.4/logs which is now configured via log4j2.properties [2017-05-16T10:27:36,957][INFO ][logstash.setting.writabledirectory] Creating directory {:setting=>"path.queue", :path=>"/home/rzxes/logstash-6.2.4/data/queue"} [2017-05-16T10:27:37,041][INFO ][logstash.agent ] No persistent UUID file found. Generating new UUID {:uuid=>"c987803c-9b18-4395-bbee-a83a90e6ea60", :path=>"/home/rzxes/logstash-6.2.4/data/uuid"} [2017-05-16T10:27:37,581][INFO ][logstash.pipeline ] Starting pipeline {"id"=>"main", "pipeline.workers"=>1, "pipeline.batch.size"=>125, "pipeline.batch.delay"=>5, "pipeline.max\_inflight"=>125} [2017-05-16T10:27:37,682][INFO ][logstash.pipeline ] Pipeline main started The stdin plugin is now waiting for input: [2017-05-16T10:27:37,886][INFO ][logstash.agent ] Successfully started Logstash API endpoint {:port=>9600}

## 二、Logstash-6.2.4连接mysql作为数据源，ES作为数据输出端：

* 由于此版本的logstash已经集成了jdbc插件，我们只需要添加一个配置文件xxx.conf。内容如下test.conf：
* input { stdin { } jdbc { # 数据库地址 端口 数据库名 jdbc\_connection\_string => "jdbc:mysql://IP:3306/dbname" # 数据库用户名 jdbc\_user => "user" # 数据库密码 jdbc\_password => "pass" # mysql java驱动地址 jdbc\_driver\_library => "/home/rzxes/logstash-6.2.4/mysql-connector-java-5.1.17.jar" jdbc\_driver\_class => "com.mysql.jdbc.Driver" jdbc\_paging\_enabled => "true" jdbc\_page\_size => "100000" # sql 语句文件，也可以直接写SQL，如statement => "select \* from table1" statement\_filepath => "/home/rzxes/logstash-6.2.4/test.sql" schedule => "\* \* \* \* \*" type => "jdbc" } } output { stdout { codec => json\_lines } elasticsearch { hosts => "192.168.230.150:9200" index => "test-1" #索引名称 document\_type => "form" #type名称 document\_id => "%{id}" #id必须是待查询的数据表的序列字段

} }

* 创建一个SQL文件：如上配置test.sql内容: select \* from table1
* test.conf,test.sql文件都在logstash的根目录下。
* 运行logstash脚本导入数据： bin/logstash -f test.conf 启动如下;



* 等待数据导入完成。开启Es-head,访问9100端口如下：



* 可以看到已经导入了11597条数据。
* 更多详细的配置参考官方文档：[plugins-inputs-jdbc-jdbc\_driver\_library](https://www.elastic.co/guide/en/logstash/5.3/plugins-inputs-jdbc.html" \l "plugins-inputs-jdbc-jdbc_driver_library)

## Ik分词器安装

由于版本要和elasticsearch相同，网上暂无，后续加上

# java代码应用

## 连接

**private static** String *host*=**"192.168.2.100"**; *// 服务器地址***private static int** *port*=9300; *// 端口***public static final** String ***CLUSTER\_NAME***=**"my-application"**; *// 集群名称*

*//新建配置，添加集群*

**private static** Settings.Builder *settings*=Settings.*builder*().put(**"cluster.name"**,***CLUSTER\_NAME***);  
  
**public static void** main(String[] args) **throws** Exception{  
 **@SuppressWarnings**({ **"resource"**, **"unchecked"** })  
 TransportClient client = **new** PreBuiltTransportClient(*settings*.build())  
 .addTransportAddress(**new** TransportAddress(InetAddress.*getByName*(*host*),*port*));  
 System.***out***.println(client);  
 client.close();  
}

以上为集群的连接操作，如果只是单机连接或者说没有setting其他的东西

TransportClient client = **new** PreBuiltTransportClient(Settings.***EMPTY***)  
 .addTransportAddress(**new** TransportAddress(InetAddress.*getByName*(*host*),*port*));

## 其他测试代码

见如下项目

