参考：<http://467754239.blog.51cto.com/4878013/1700828/>

**ELK 日志分析系统**

2015-10-08 13:18:58

标签：[elk](http://blog.51cto.com/tag-elk.html)

原创作品，允许转载，转载时请务必以超链接形式标明文章 [原始出处](http://467754239.blog.51cto.com/4878013/1700828) 、作者信息和本声明。否则将追究法律责任。<http://467754239.blog.51cto.com/4878013/1700828>

# 大纲

一、简介

二、Logstash

三、Redis

四、Elasticsearch

五、Kibana

# 一、简介

## 1、核心组成

[ELK](http://467754239.blog.51cto.com/4878013/1700828)由Elasticsearch、Logstash和Kibana三部分组件组成；

### Elastic

Elasticsearch是个开源分布式搜索引擎，它的特点有：分布式，零配置，自动发现，索引自动分片，索引副本机制，restful风格接口，多数据源，自动搜索负载等。

### Logstash

Logstash是一个完全开源的工具，它可以对你的日志进行收集、分析，并将其存储供以后使用(me：类似数据清洗的功能)

### Kibabna

kibana 是一个开源和免费的工具，它可以为 Logstash 和 ElasticSearch 提供的日志分析友好的 Web 界面，可以帮助您汇总、分析和搜索重要数据日志。

## 2、四大组件

Logstash: logstash server端用来搜集日志；

Elasticsearch: 存储各类日志；

Kibana: web化接口用作查寻和可视化日志；

Logstash Forwarder: logstash client端用来通过lumberjack 网络协议发送日志到logstash server；

## 3、ELK工作流程

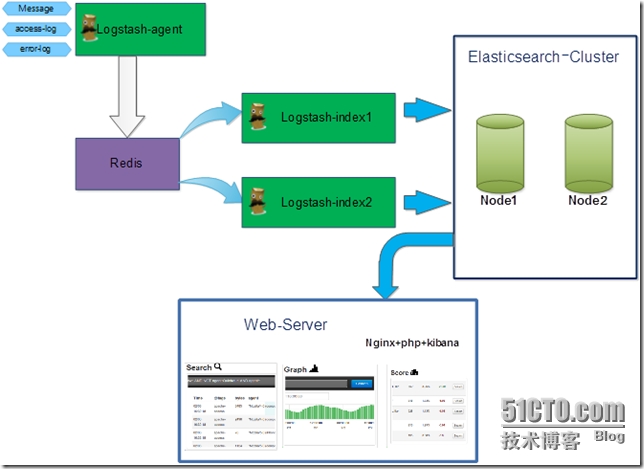
在需要收集日志的所有服务上部署logstash，作为logstash agent（logstash shipper）用于监控并过滤收集日志，将过滤后的内容发送到Redis，然后logstash indexer将日志收集在一起交给全文搜索服务ElasticSearch，可以用ElasticSearch进行自定义搜索通过Kibana 来结合自定义搜索进行页面展示。

Logstash收集数据

Elasticsearch 存储数据

Kibana展示数据

Redis缓存

[](http://s3.51cto.com/wyfs02/M01/74/25/wKioL1YWGBnxpAwyAAGFWRpT6X8070.jpg)

## 4、ELK的帮助手册

ELK官网：https://www.elastic.co/

ELK官网文档：https://www.elastic.co/guide/index.html

ELK中文手册：http://kibana.logstash.es/content/elasticsearch/monitor/logging.html

注释

ELK有两种安装方式

(1)集成环境：Logstash有一个集成包，里面包括了其全套的三个组件；也就是安装一个集成包。

(2)独立环境：三个组件分别单独安装、运行、各司其职。(比较常用)

本实验也以第二种方式独立环境来进行演示；单机版主机地址为：192.168.1.104

# 二、Logstash

### 1、安装jdk

|  |
| --- |
| Logstash的运行依赖于Java运行环境。  # yum -y install java-1.8.0  # java –version  openjdk version "1.8.0\_51"  OpenJDK Runtime Environment (build 1.8.0\_51-b16)  OpenJDK 64-Bit Server VM (build 25.51-b03, mixed mode) |

### 2、安装logstash

|  |
| --- |
| # wget https://download.elastic.co/logstash/logstash/logstash-1.5.4.tar.gz  # tar zxf logstash-1.5.4.tar.gz -C /usr/local/  配置logstash的环境变量  vi /etc/profile.d/logstash.sh  export LOGSTASH\_HOME=/home/~/logstash-x.x.x  export PATH=$PATH:$LOGSTASH\_HOME/bin  保存后使用命令使其生效：  source /etc/profile.d/logstash.sh |

### 3、logstash常用参数

|  |
| --- |
| -e :指定logstash的配置信息，可以用于快速测试;  -f :指定logstash的配置文件；可以用于生产环境; |

### 4、启动logstash

### 4.1 通过-e参数指定logstash的配置信息，用于快速测试，直接输出到屏幕。

|  |
| --- |
| [hadoop@master logstash-5.5.0]$ logstash -e "input {stdin{}} output {stdout{}}"  ERROR StatusLogger No log4j2 configuration file found. Using default configuration: logging only errors to the console.  Sending Logstash's logs to /home/hadoop/elk/logstash-5.5.0/logs which is now configured via log4j2.properties  [2017-07-20T07:16:25,659][INFO ][logstash.agent ] No persistent UUID file found. Generating new UUID {:uuid=>"dd45edc9-39d2-4037-8cc1-e77b152c959f", :path=>"/home/hadoop/elk/logstash-5.5.0/data/uuid"}  [2017-07-20T07:16:26,446][INFO ][logstash.pipeline ] Starting pipeline {"id"=>"main", "pipeline.workers"=>1, "pipeline.batch.size"=>125, "pipeline.batch.delay"=>5, "pipeline.max\_inflight"=>125}  [2017-07-20T07:16:26,555][INFO ][logstash.pipeline ] Pipeline main started  The stdin plugin is now waiting for input:  这时进入到交互模式：输入一句然后以特定的日志格式输出一句  my name is Daniel //手动输入后回车，等待10秒后会有返回结果  2015-10-08T13:55:50.660Z 0.0.0.0 my name is Daniel.  hello  2017-07-20T14:17:41.175Z master hello //master是因为计算机名字为master  the site is sina  2017-07-20T14:17:52.954Z master the site is sina |

### .2 通过-e参数指定logstash的配置信息，用于快速测试，以json格式输出到屏幕。

|  |
| --- |
| [hadoop@master config]$ logstash -e 'input{stdin{}} output{stdout{codec=>rubydebug}}'  hello  2017-07-20 07:27:41,883 main ERROR Unable to locate appender "${sys:ls.log.format}\_console" for logger config "root"  2017-07-20 07:27:41,884 main ERROR Unable to locate appender "${sys:ls.log.format}\_rolling" for logger config "root"  2017-07-20 07:27:41,885 main ERROR Unable to locate appender "${sys:ls.log.format}\_rolling\_slowlog" for logger config "slowlog"  2017-07-20 07:27:41,889 main ERROR Unable to locate appender "${sys:ls.log.format}\_console\_slowlog" for logger config "slowlog"  Sending Logstash's logs to /home/hadoop/elk/logstash-5.5.0/logs which is now configured via log4j2.properties  [2017-07-20T07:27:50,722][INFO ][logstash.pipeline ] Starting pipeline {"id"=>"main", "pipeline.workers"=>1, "pipeline.batch.size"=>125, "pipeline.batch.delay"=>5, "pipeline.max\_inflight"=>125}  [2017-07-20T07:27:50,787][INFO ][logstash.pipeline ] Pipeline main started  The stdin plugin is now waiting for input:  {  "@timestamp" => 2017-07-20T14:27:50.912Z,  "@version" => "1",  "host" => "master",  "message" => "hello"  }  [2017-07-20T07:27:51,030][INFO ][logstash.agent ] Successfully started Logstash API endpoint {:port=>9600}  hello  {  "@timestamp" => 2017-07-20T14:27:57.728Z,  "@version" => "1",  "host" => "master",  "message" => "hello"  }  this is sina  {  "@timestamp" => 2017-07-20T14:28:01.981Z,  "@version" => "1",  "host" => "master",  "message" => "this is sina"  } |

### 5、logstash以配置文件方式启动

## 5.1 输出信息到屏幕

|  |
| --- |
| # vim logstash-simple.conf  input { stdin {} }  output {     stdout { codec=> rubydebug }  }    [hadoop@master config]$ logstash -f ../config/logstash-simple.conf  hello  2017-07-20 07:29:54,682 main ERROR Unable to locate appender "${sys:ls.log.format}\_console" for logger config "root"  2017-07-20 07:29:54,683 main ERROR Unable to locate appender "${sys:ls.log.format}\_rolling" for logger config "root"  2017-07-20 07:29:54,685 main ERROR Unable to locate appender "${sys:ls.log.format}\_rolling\_slowlog" for logger config "slowlog"  2017-07-20 07:29:54,691 main ERROR Unable to locate appender "${sys:ls.log.format}\_console\_slowlog" for logger config "slowlog"  Sending Logstash's logs to /home/hadoop/elk/logstash-5.5.0/logs which is now configured via log4j2.properties  [2017-07-20T07:29:58,786][INFO ][logstash.pipeline ] Starting pipeline {"id"=>"main", "pipeline.workers"=>1, "pipeline.batch.size"=>125, "pipeline.batch.delay"=>5, "pipeline.max\_inflight"=>125}  [2017-07-20T07:29:58,865][INFO ][logstash.pipeline ] Pipeline main started  The stdin plugin is now waiting for input:  {  "@timestamp" => 2017-07-20T14:29:59.020Z,  "@version" => "1",  "host" => "master",  "message" => "hello"  }  [2017-07-20T07:29:59,117][INFO ][logstash.agent ] Successfully started Logstash API endpoint {:port=>9600}  dd  {  "@timestamp" => 2017-07-20T14:30:02.526Z,  "@version" => "1",  "host" => "master",  "message" => "dd"  } |

### 5.2 logstash输出信息存储到redis数据库中

刚才我们是将信息直接显示在屏幕上了，现在我们将logstash的输出信息保存到redis数据库中,如下

|  |
| --- |
| 前提是本地(192.168.1.104)有redis数据库，那么下一步我们就是安装redis数据库.  # cat logstash\_to\_redis.conf  input { stdin { } }  output {      stdout { codec => rubydebug }      redis {          host => '192.168.1.104'          data\_type => 'list'          key => 'logstash:redis'      }  }    如果提示Failed to send event to Redis，表示连接Redis失败或者没有安装，请检查... |

### 6、 查看logstash的监听端口号

|  |
| --- |
| # logstash -f logstash\_to\_redis.conf --verbose  # netstat -tnlp |grep java  tcp        0      0 :::9301                     :::\*                        LISTEN      1326/java |

# 三、Redis

## 1、安装Redis

|  |
| --- |
| wget http://download.redis.io/releases/redis-2.8.19.tar.gz  yum install tcl -y  tar zxf redis-2.8.19.tar.gz  cd redis-2.8.19  make MALLOC=libc  make test    //这一步时间会稍久点...  make install    cd utils/  ./install\_server.sh     //脚本执行后，所有选项都以默认参数为准即可  Welcome to the redis service installer  This script will help you easily set up a running redis server    Please select the redis port for this instance: [6379]  Selecting default: 6379  Please select the redis config file name [/etc/redis/6379.conf]  Selected default - /etc/redis/6379.conf  Please select the redis log file name [/var/log/redis\_6379.log]  Selected default - /var/log/redis\_6379.log  Please select the data directory for this instance [/var/lib/redis/6379]  Selected default - /var/lib/redis/6379  Please select the redis executable path [/usr/local/bin/redis-server]  Selected config:  Port           : 6379  Config file    : /etc/redis/6379.conf  Log file       : /var/log/redis\_6379.log  Data dir       : /var/lib/redis/6379  Executable     : /usr/local/bin/redis-server  Cli Executable : /usr/local/bin/redis-cli  Is this ok? Then press ENTER to go on or Ctrl-C to abort.  Copied /tmp/6379.conf => /etc/init.d/redis\_6379  Installing service...  Successfully added to chkconfig!  Successfully added to runlevels 345!  Starting Redis server...  Installation successful! |

## 2、查看redis的监控端口

|  |
| --- |
| # netstat -tnlp |grep redis  tcp        0      0 0.0.0.0:6379                0.0.0.0:\*                   LISTEN      3843/redis-server \*  tcp        0      0 127.0.0.1:21365             0.0.0.0:\*                   LISTEN      2290/src/redis-serv  tcp        0      0 :::6379                     :::\*                        LISTEN      3843/redis-server \* |

## 3、测试redis是否正常工作

|  |
| --- |
| # cd redis-2.8.19/src/  # ./redis-cli -h 192.168.1.104 -p 6379 //连接redis  192.168.1.104:6379> ping  PONG  192.168.1.104:6379> set name zhengyansheng  OK  192.168.1.104:6379> get name  "zhengyansheng"  192.168.1.104:6379> quit |

4、redis服务启动命令

|  |
| --- |
| # ps -ef |grep redis  root      3963     1  0 08:42 ?        00:00:00 /usr/local/bin/redis-server \*:6379 |

5、redis的动态监控

|  |
| --- |
| # cd redis-2.8.19/src/  # ./redis-cli monitor     //reids动态监控 |

6、logstash结合redis工作

6.1 首先确认redis服务是启动的

|  |
| --- |
| # netstat -tnlp |grep redis  tcp        0      0 0.0.0.0:6379                0.0.0.0:\*                   LISTEN      3843/redis-server \*  tcp        0      0 127.0.0.1:21365             0.0.0.0:\*                   LISTEN      2290/src/redis-serv  tcp        0      0 :::6379                     :::\*                        LISTEN      3843/redis-server \* |

6.2 启动redis动态监控

|  |
| --- |
| # cd redis-2.8.19/src/  # ./redis-cli monitor  OK |

### 6.3 基于入口redis启动logstash

|  |
| --- |
| # cat logstash\_to\_redis.conf  input { stdin { } }  output {      stdout { codec => rubydebug }      redis {          host => '192.168.1.104'          data\_type => 'list'          key => 'logstash:redis'      }  }  # logstash agent -f logstash\_to\_redis.conf --verbose  Pipeline started {:level=>:info}  Logstash startup completed  dajihao linux  {         "message" => "dajihao linux",        "@version" => "1",      "@timestamp" => "2015-10-08T14:42:07.550Z",            "host" => "0.0.0.0"  } |

### 6.4 查看redis的监控接口上的输出

|  |  |
| --- | --- |
| 1  2  3  4  5 | # ./redis-cli monitor  OK  1444315328.103928 [0 192.168.1.104:56211] "rpush" "logstash:redis" "{\"message\":\"dajihao linux\",\"@version\":\"1\",\"@timestamp\":\"2015-10-08T14:42:07.550Z\",\"host\":\"0.0.0.0\"}"    如果redis的监控上也有以上信息输出，表明logstash和redis的结合是正常的。 |

# 四、Elasticsearch

## 1、安装Elasticsearch

|  |
| --- |
| # wget https://download.elastic.co/elasticsearch/elasticsearch/elasticsearch-1.7.2.tar.gz  # tar zxf elasticsearch-1.7.2.tar.gz -C /usr/local/ |

2、修改elasticsearch配置文件elasticsearch.yml并且做以下修改.

|  |
| --- |
| # vim /usr/local/elasticsearch-1.7.2/config/elasticsearch.yml  discovery.zen.ping.multicast.enabled: false        #关闭广播，如果局域网有机器开9300 端口，服务会启动不了  network.host: 192.168.1.104    #指定主机地址，其实是可选的，但是最好指定因为后面跟kibana集成的时候会报http连接出错（直观体现好像是监听了:::9200 而不是0.0.0.0:9200）  http.cors.allow-origin: "/.\*/"  http.cors.enabled: true        #这2项都是解决跟kibana集成的问题，错误体现是 你的 elasticsearch 版本过低，其实不是 |

## 3、启动elasticsearch服务

|  |  |
| --- | --- |
| 1  2  3 | # /usr/local/elasticsearch-1.7.2/bin/elasticsearch     #日志会输出到stdout  # /usr/local/elasticsearch-1.7.2/bin/elasticsearch -d #表示以daemon的方式启动  # nohup /usr/local/elasticsearch-1.7.2/bin/elasticsearch > /var/log/logstash.log 2>&1 & |

## 4、查看elasticsearch的监听端口

|  |
| --- |
| # netstat -tnlp |grep java  tcp        0      0 :::9200                     :::\*                        LISTEN      7407/java  tcp        0      0 :::9300                     :::\*                        LISTEN      7407/java |

## 5、elasticsearch和logstash结合

|  |
| --- |
| 将logstash的信息输出到elasticsearch中  # cat logstash-elasticsearch.conf  input { stdin {} }  output {      elasticsearch { host => "192.168.1.104" }      stdout { codec=> rubydebug }  } |

## 6、基于配置文件启动logstash

|  |
| --- |
| # /usr/local/logstash-1.5.4/bin/logstash -f logstash-elasticsearch.conf  Pipeline started {:level=>:info}  Logstash startup completed  python linux java c++    //手动输入  {     "message" => "python linux java c++",     "@version" => "1",     "@timestamp" => "2015-10-08T14:51:56.899Z",     "host" => "0.0.0.0"  } |

## 7、curl命令发送请求来查看elasticsearch是否接收到了数据

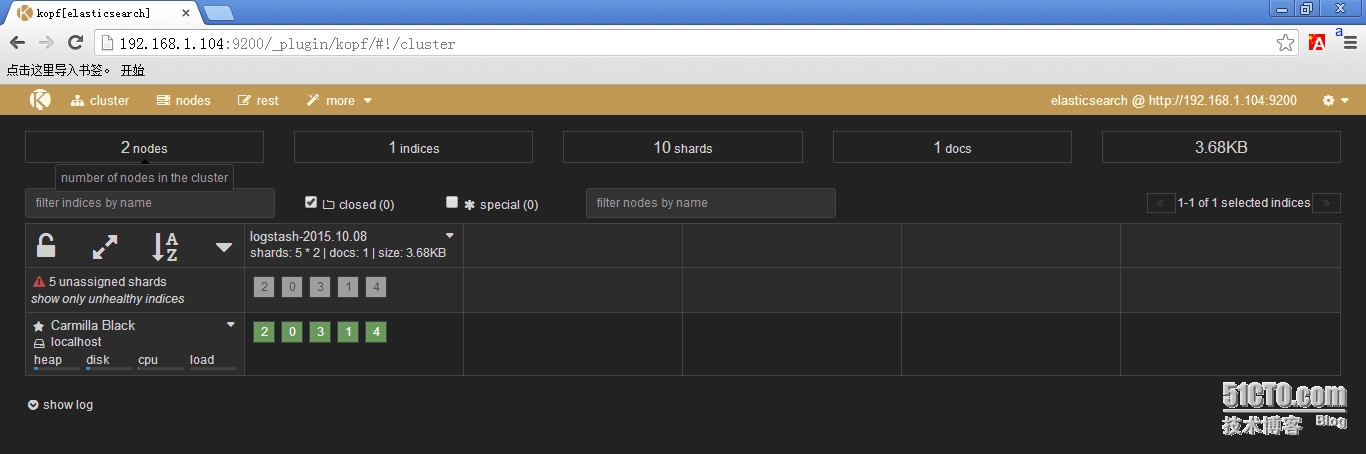
|  |
| --- |
| # curl http://localhost:9200/\_search?pretty  {    "took" : 28,    "timed\_out" : false,    "\_shards" : {      "total" : 5,      "successful" : 5,      "failed" : 0    },    "hits" : {      "total" : 1,      "max\_score" : 1.0,      "hits" : [ {        "\_index" : "logstash-2015.10.08",        "\_type" : "logs",        "\_id" : "AVBH7-6MOwimSJSPcXjb",        "\_score" : 1.0,        "\_source":{"message":"python linux java c++","@version":"1","@timestamp":"2015-10-08T14:51:56.899Z","host":"0.0.0.0"}      } ]    }  } |

## 8、安装elasticsearch插件

|  |
| --- |
| #Elasticsearch-kopf插件可以查询Elasticsearch中的数据，安装elasticsearch-kopf，只要在你安装Elasticsearch的目录中执行以下命令即可：  # cd /usr/local/elasticsearch-1.7.2/bin/  # ./plugin install lmenezes/elasticsearch-kopf  -> Installing lmenezes/elasticsearch-kopf...  Trying https://github.com/lmenezes/elasticsearch-kopf/archive/master.zip...  Downloading .............................................................................................  Installed lmenezes/elasticsearch-kopf into /usr/local/elasticsearch-1.7.2/plugins/kopf    执行插件安装后会提示失败，很有可能是网络等情况...  -> Installing lmenezes/elasticsearch-kopf...  Trying https://github.com/lmenezes/elasticsearch-kopf/archive/master.zip...  Failed to install lmenezes/elasticsearch-kopf, reason: failed to download out of all possible locations..., use --verbose to get detailed information    解决办法就是手动下载该软件，不通过插件安装命令...  cd /usr/local/elasticsearch-1.7.2/plugins  wget https://github.com/lmenezes/elasticsearch-kopf/archive/master.zip  unzip master.zip  mv elasticsearch-kopf-master kopf  以上操作就完全等价于插件的安装命令 |

## 9、浏览器访问kopf页面访问elasticsearch保存的数据

|  |
| --- |
| # netstat -tnlp |grep java  tcp        0      0 :::9200                     :::\*                        LISTEN      7969/java  tcp        0      0 :::9300                     :::\*                        LISTEN      7969/java  tcp        0      0 :::9301                     :::\*                        LISTEN      8015/java |

[](http://s3.51cto.com/wyfs02/M01/74/27/wKiom1YWE7GDnjRIAALZuJ3R5GM868.jpg)

## 10、从redis数据库中读取然后输出到elasticsearch中

|  |
| --- |
| # cat logstash-redis.conf  input {      redis {          host => '192.168.1.104'  # 我方便测试没有指定password，最好指定password          data\_type => 'list'          port => "6379"          key => 'logstash:redis' #自定义          type => 'redis-input'   #自定义      }  }  output {      elasticsearch {          host => "192.168.1.104"          codec => "json"          protocol => "http"  #版本1.0+ 必须指定协议http      }  } |

# 五、Kinana

## 1、安装Kibana

|  |
| --- |
| # wget https://download.elastic.co/kibana/kibana/kibana-4.1.2-linux-x64.tar.gz  # tar zxf kibana-4.1.2-linux-x64.tar.gz -C /usr/local |

## 2、修改kibana配置文件kibana.yml

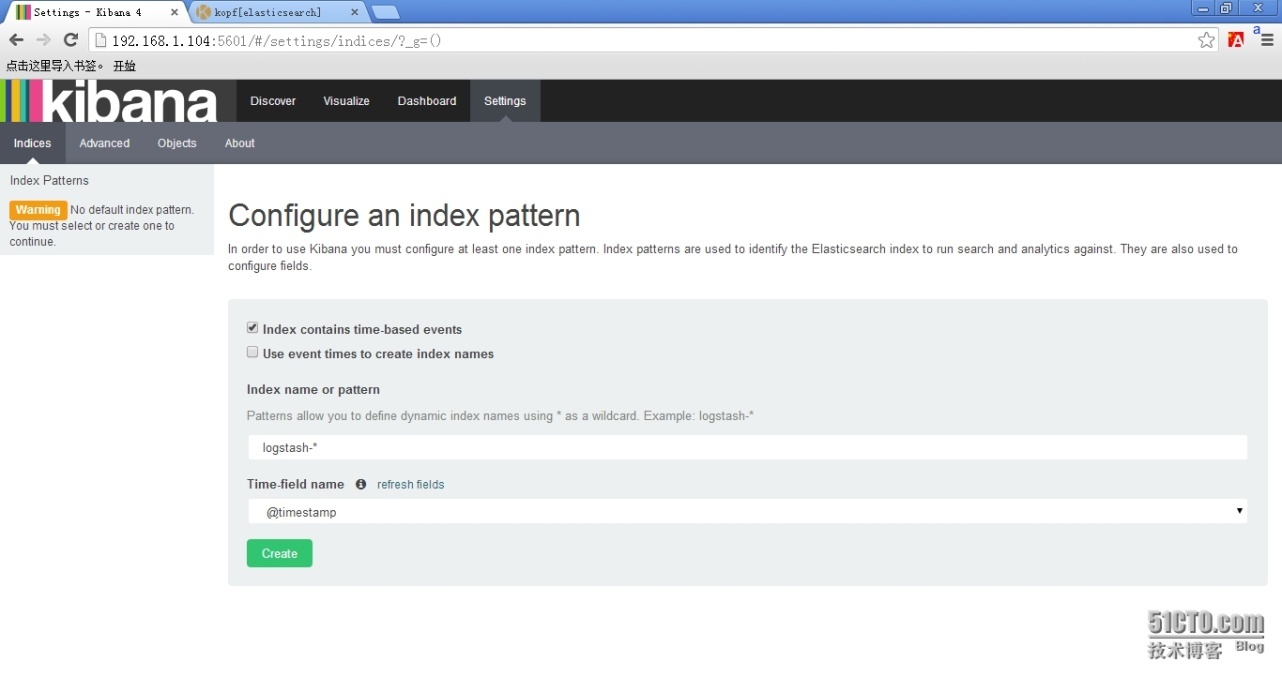
|  |  |
| --- | --- |
| 1  2 | # vim /usr/local/kibana-4.1.2-linux-x64/config/kibana.yml  elasticsearch\_url: "http://192.168.1.104:9200" |

## 3、启动kibana

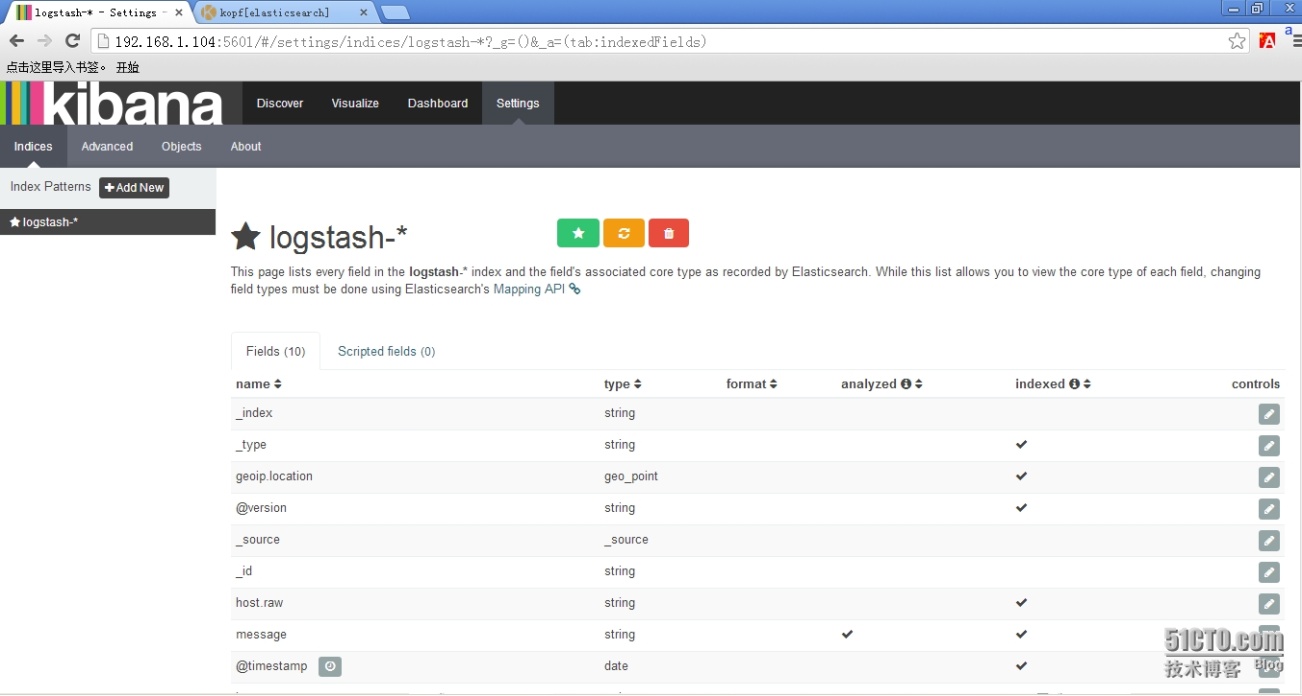
|  |
| --- |
| /usr/local/kibana-4.1.2-linux-x64/bin/kibana    输出以下信息，表明kibana成功.  {"name":"Kibana","hostname":"localhost.localdomain","pid":1943,"level":30,"msg":"No existing kibana index found","time":"2015-10-08T00:39:21.617Z","v":0}  {"name":"Kibana","hostname":"localhost.localdomain","pid":1943,"level":30,"msg":"Listening on 0.0.0.0:5601","time":"2015-10-08T00:39:21.637Z","v":0}  kibana默认监听在本地的5601端口上 |

## 4、浏览器访问kibana

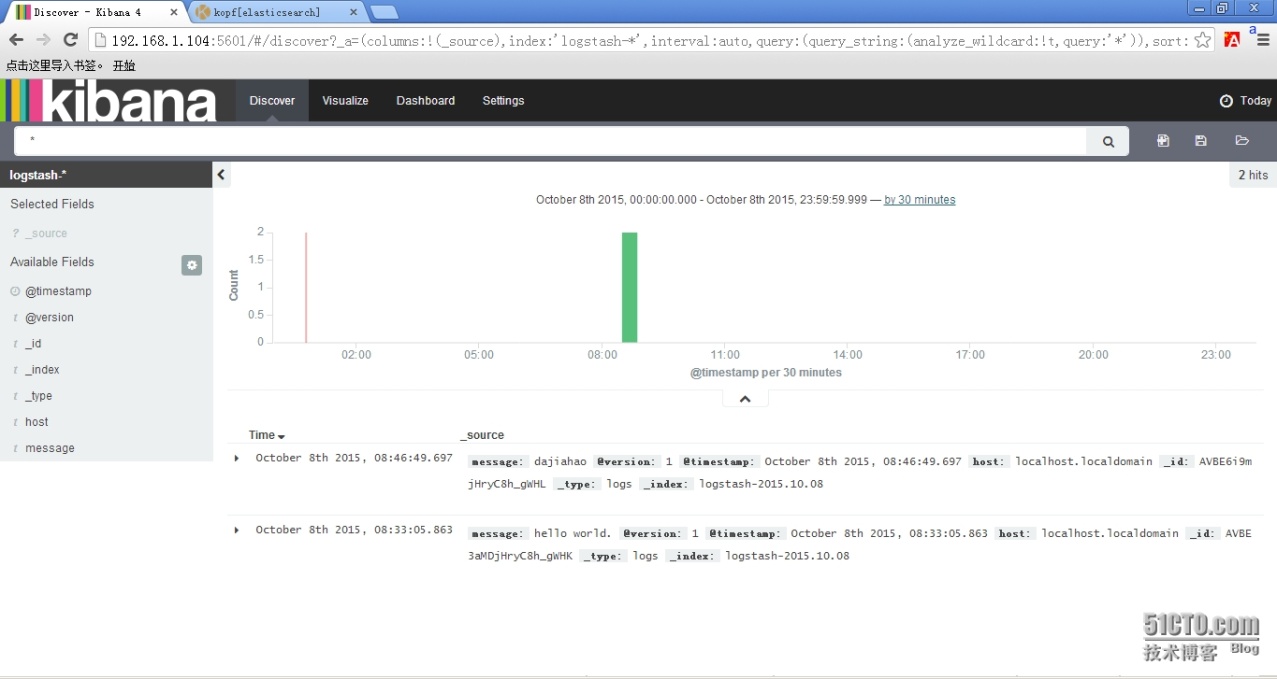
### 4.1 使用默认的logstash-\*的索引名称，并且是基于时间的，点击“Create”即可。

[](http://s3.51cto.com/wyfs02/M02/74/25/wKioL1YWFTbx56DuAAOecFQkxcA301.jpg)

### 4.2 看到如下界面说明索引创建完成。

[](http://s3.51cto.com/wyfs02/M00/74/27/wKiom1YWFR-hbgxOAAQZ85RcxMg067.jpg)

### 4.3 点击“Discover”，可以搜索和浏览Elasticsearch中的数据。

[](http://s3.51cto.com/wyfs02/M00/74/25/wKioL1YWFTaRWGZmAAQ81v02YE8823.jpg)

# >>>结束<<<

|  |
| --- |
| 1、ELK默认端口号  elasticsearch：9200 9300  logstash     : 9301  kibana       : 5601    2、错误汇总  (1)java版本过低  [2015-10-07 18:39:18.071]  WARN -- Concurrent: [DEPRECATED] Java 7 is deprecated, please use Java 8.    (2)Kibana提示Elasticsearch版本过低...  This version of Kibana requires Elasticsearch 2.0.0 or higher on all nodes. I found the following incompatible nodes in your cluster:  Elasticsearch v1.7.2 @ inet[/192.168.1.104:9200] (127.0.0.1)  解决办法：  Redis连接不上也可能是redis的服务器版本和logstash内置的client的版本不一致 |

软件包以打包上传：<http://pan.baidu.com/s/1hqfeFvY>