TTS 10.0 COOKBOOK

(NSD ARCHITECTURE DAY02)

版本编号 10.0

2018-08

达内 IT 培训集团

NSD ARCHITECTURE DAY04

1. 案例1：导入数据

- **问题**

本案例要求批量导入数据：

  - 批量导入数据并查看

- **步骤**

实现此案例需要按照如下步骤进行。

步骤一：导入数据

1）使用_bulk 批量导入数据

使用 POST 方式批量导入数据，数据格式为 json，url 编码使用 data-binary 导入含有 index 配置的 json 文件

```
[root@room9pc01 ~]# scp /var/ftp/elk/*.gz 192.168.1.66:/root/
[root@kibana ~]# gzip  -d logs.jsonl.gz
[root@kibana ~]#  gzip  -d accounts.json.gz
[root@kibana ~]# gzip  -d shakespeare.json.gz
[root@kibana ~]# curl -X POST "http://192.168.1.61:9200/_bulk" \
--data-binary @shakespeare.json
[root@kibana ~]# curl -X POST "http://192.168.1.61:9200/xixi/haha/_bulk" \
 --data-binary @accounts.json
//索引是 xixi，类型是 haha，必须导入索引和类型，没有索引，要加上
[root@kibana ~]# curl -X POST "http://192.168.1.61:9200/_bulk"  \
--data-binary @logs.jsonl
```

2）使用 GET 查询结果

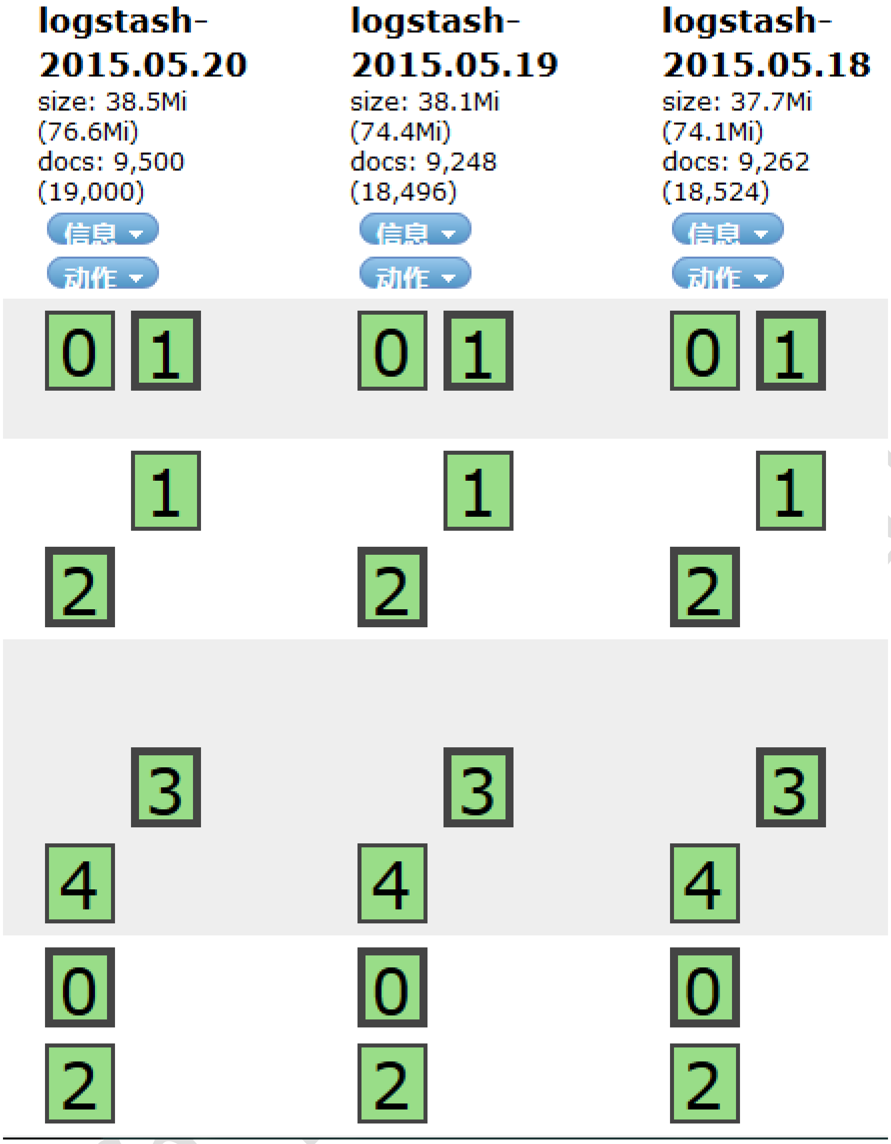
```
[root@kibana ~]# curl -XGET 'http://192.168.1.61:9200/_mget?pretty' -d '{
 "docs":[
     {
        "_index":"shakespeare",
        "_type:":"act",
        "_id":0
},
{
        "_index":"shakespeare",
        "_type:":"line",
        "_id":0
},
{
        "_index":"xixi",
        "_type:":"haha",
        "_id":25
}
]
}'
```

```
{         //查询的结果
  "docs" : [ {
    "_index" : "shakespeare",
    "_type" : "act",
    "_id" : "0",
    "_version" : 1,
    "found" : true,
    "_source" : {
      "line_id" : 1,
      "play_name" : "Henry IV",
      "speech_number" : "",
      "line_number" : "",
      "speaker" : "",
      "text_entry" : "ACT I"
    }
  }, {
    "_index" : "shakespeare",
    "_type" : "act",
    "_id" : "0",
    "_version" : 1,
    "found" : true,
    "_source" : {
      "line_id" : 1,
      "play_name" : "Henry IV",
      "speech_number" : "",
      "line_number" : "",
      "speaker" : "",
      "text_entry" : "ACT I"
    }
  }, {
    "_index" : "xixi",
    "_type" : "haha",
    "_id" : "25",
    "_version" : 1,
    "found" : true,
    "_source" : {
      "account_number" : 25,
      "balance" : 40540,
      "firstname" : "Virginia",
      "lastname" : "Ayala",
      "age" : 39,
      "gender" : "F",
      "address" : "171 Putnam Avenue",
      "employer" : "Filodyne",
      "email" : "virginiaayala@filodyne.com",
      "city" : "Nicholson",
      "state" : "PA"
    }
  } ]
}
```

步骤二：使用 kibana 查看数据是否导入成功

1）数据导入以后查看 logs 是否导入成功，如图-1 所示：

```
[root@se5 ~]# firefox http://192.168.1.65:9200/_plugin/head/
```



logstash-2015.05.20
size: 38.5Mi
(76.6Mi)
docs: 9,500
(19,000)

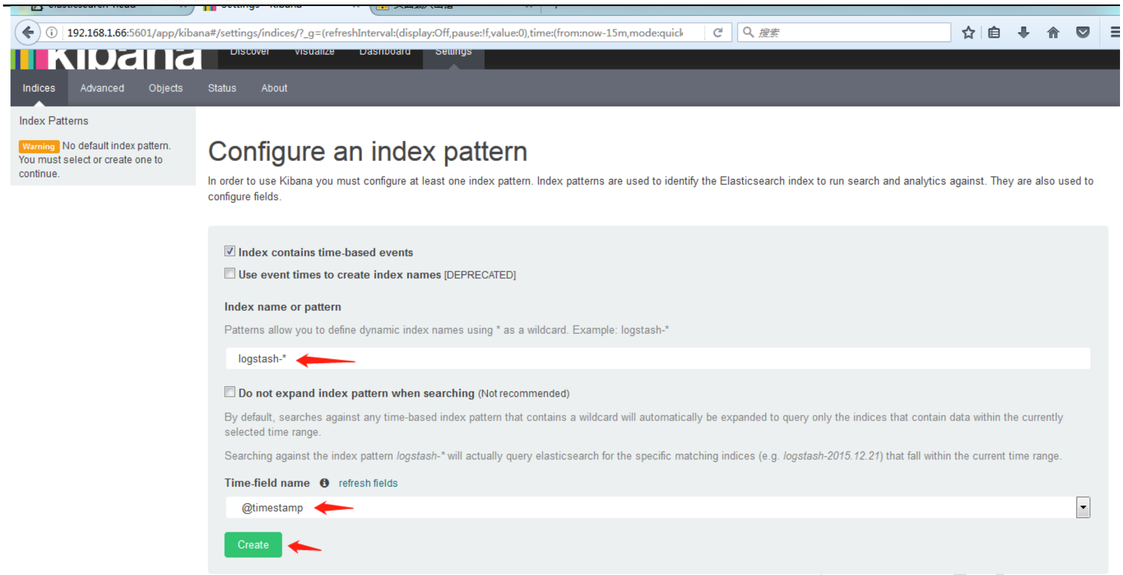
logstash-2015.05.19
size: 38.1Mi
(74.4Mi)
docs: 9,248
(18,496)

logstash-2015.05.18
size: 37.7Mi
(74.1Mi)
docs: 9,262
(18,524)

图-1

2）kibana 导入数据，如图-2 所示：

```
[root@kibana ~]# firefox  http://192.168.1.66:5601
```



192.168.1.66:5601/app/kibana#/settings/indices/?_g=(refreshInterval:(display:Off,pause:!f,value:0),time:(from:now-15m,mode:quick

Warning No default index pattern. You must select or create one to continue.

Configure an index pattern

In order to use Kibana you must configure at least one index pattern. Index patterns are used to identify the Elasticsearch index to run search and analytics against. They are also used to configure fields.

☑ Index contains time-based events

□ Use event times to create index names [DEPRECATED]

Index name or pattern

Patterns allow you to define dynamic index names using * as a wildcard. Example: logstash-*

logstash-*

□ Do not expand index pattern when searching (Not recommended)

By default, searches against any time-based index pattern that contains a wildcard will automatically be expanded to query only the indices that contain data within the currently selected time range.

Searching against the index pattern *logstash-** will actually query elasticsearch for the specific matching indices (e.g. *logstash-2015.12.21*) that fall within the current time range.

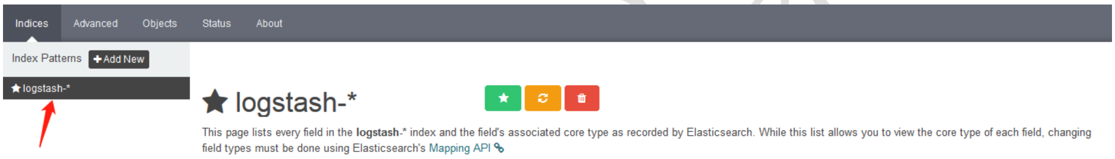
Time-field name refresh fields

@timestamp

Create

图-2

3）成功创建会有 logstash-*，如图-3 所示：

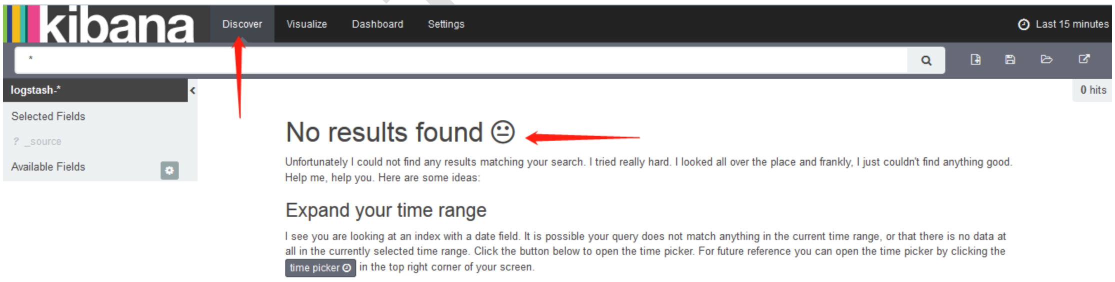


★ logstash-*

This page lists every field in the **logstash-*** index and the field's associated core type as recorded by Elasticsearch. While this list allows you to view the core type of each field, changing field types must be done using Elasticsearch's Mapping API

图-3

4）导入成功之后选择 Discover，如图-4 所示：



No results found 😐

Unfortunately I could not find any results matching your search. I tried really hard. I looked all over the place and frankly, I just couldn't find anything good. Help me, help you. Here are some ideas:

Expand your time range

I see you are looking at an index with a date field. It is possible your query does not match anything in the current time range, or that there is no data at all in the currently selected time range. Click the button below to open the time picker. For future reference you can open the time picker by clicking the time picker in the top right corner of your screen.

图-4

注意： 这里没有数据的原因是导入日志的时间段不对，默认配置是最近 15 分钟，在这可以修改一下时间来显示

5）kibana 修改时间，选择 Lsat 15 miuntes，如图-5 所示：

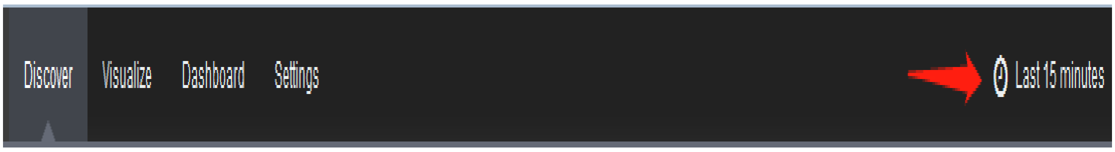


图-5

6）选择 Absolute，如图-6 所示：

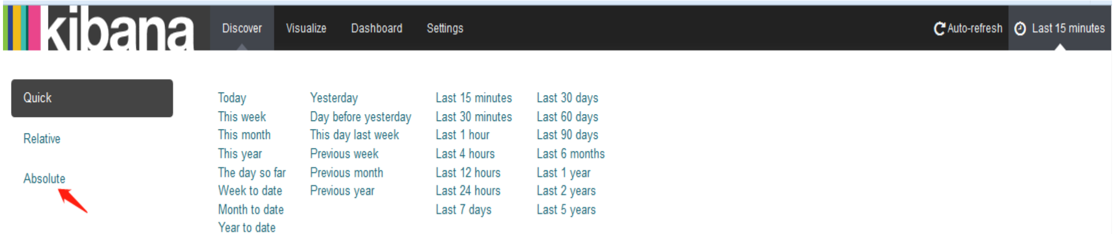
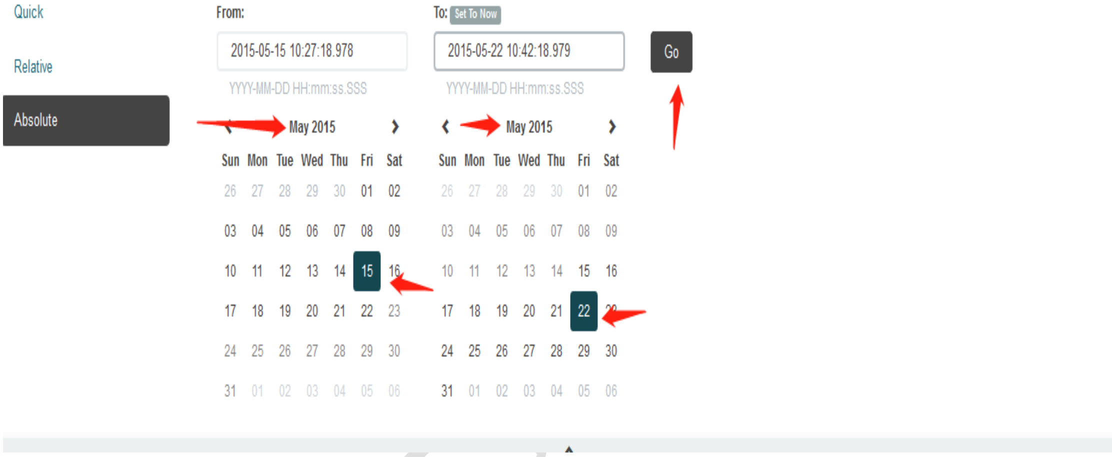


图-6

7）选择时间 2015-5-15 到 2015-5-22，如图-7 所示：



From: 2015-05-15 10:27:18.978

To: 2015-05-22 10:42:18.979

图-7

8）查看结果，如图-8 所示：

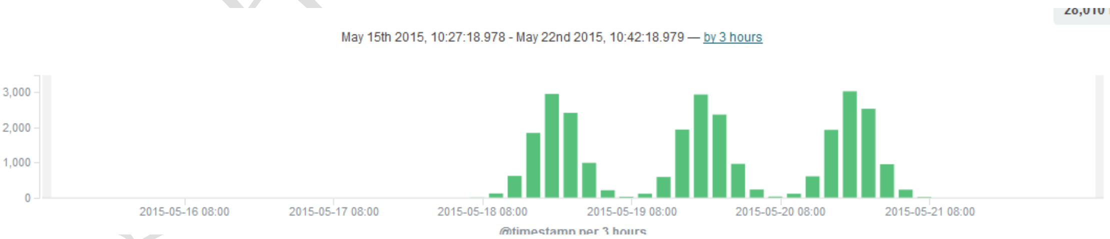
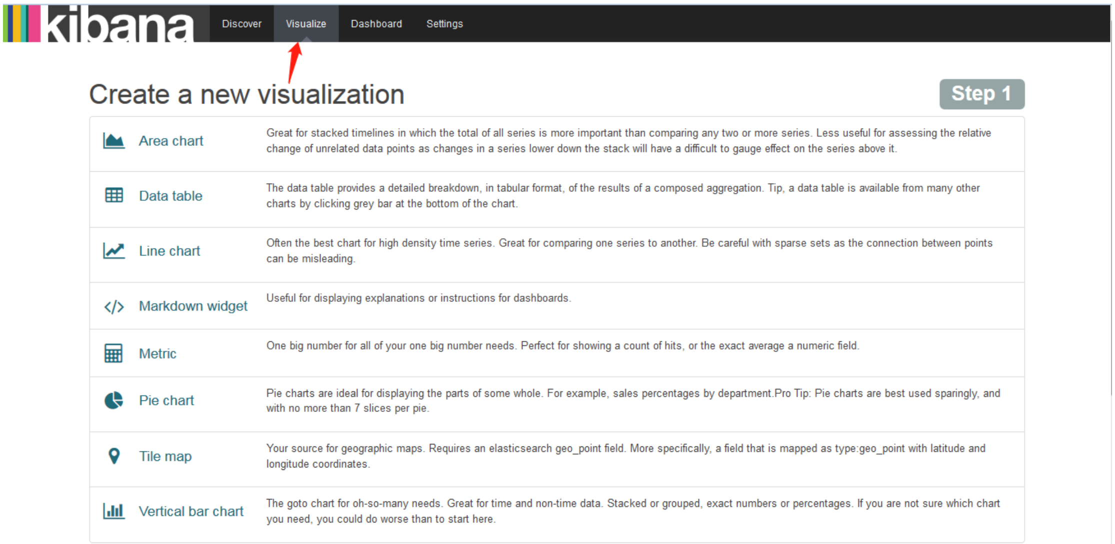

May 15th 2015, 10:27:18.978 - May 22nd 2015, 10:42:18.979 — by 3 hours

图-8

9）除了柱状图，Kibana 还支持很多种展示方式 ，如图-9 所示：

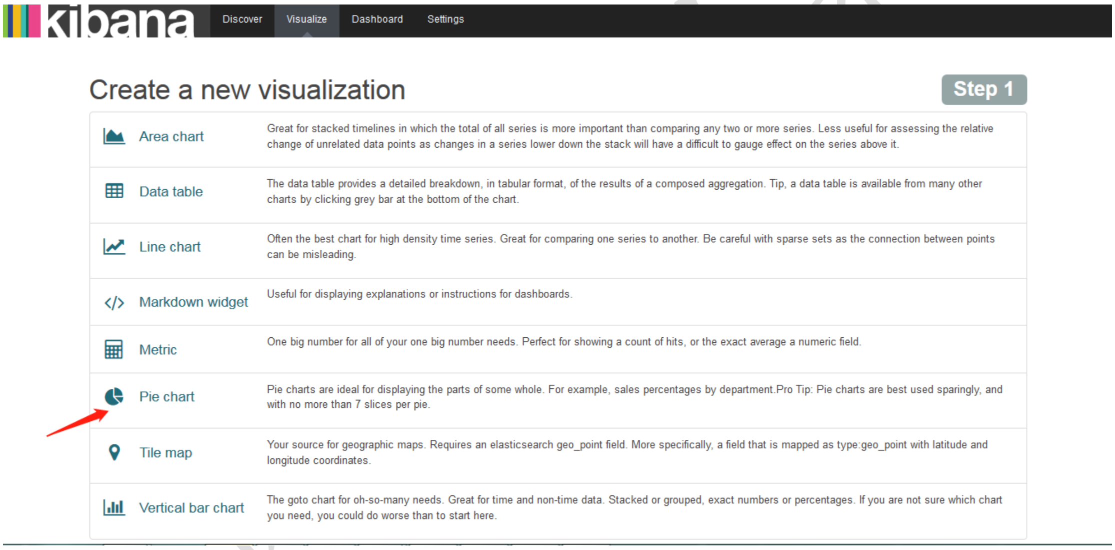


Create a new visualization

Area chart	Great for stacked timelines in which the total of all series is more important than comparing any two or more series. Less useful for assessing the relative change of unrelated data points as changes in a series lower down the stack will have a difficult to gauge effect on the series above it.
Data table	The data table provides a detailed breakdown, in tabular format, of the results of a composed aggregation. Tip, a data table is available from many other charts by clicking grey bar at the bottom of the chart.
Line chart	Often the best chart for high density time series. Great for comparing one series to another. Be careful with sparse sets as the connection between points can be misleading.
Markdown widget	Useful for displaying explanations or instructions for dashboards.
Metric	One big number for all of your one big number needs. Perfect for showing a count of hits, or the exact average a numeric field.
Pie chart	Pie charts are ideal for displaying the parts of some whole. For example, sales percentages by department.Pro Tip: Pie charts are best used sparingly, and with no more than 7 slices per pie.
Tile map	Your source for geographic maps. Requires an elasticsearch geo_point field. More specifically, a field that is mapped as type:geo_point with latitude and longitude coordinates.
Vertical bar chart	The goto chart for oh-so-many needs. Great for time and non-time data. Stacked or grouped, exact numbers or percentages. If you are not sure which chart you need, you could do worse than to start here.

图-9

10）做一个饼图，选择 Pie chart，如图-10 所示：

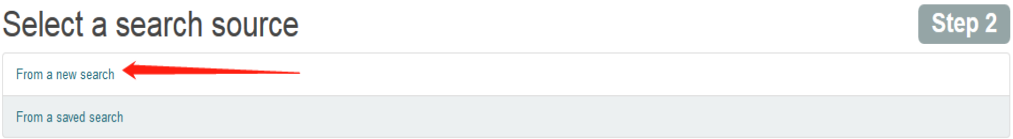


Create a new visualization

Area chart	Great for stacked timelines in which the total of all series is more important than comparing any two or more series. Less useful for assessing the relative change of unrelated data points as changes in a series lower down the stack will have a difficult to gauge effect on the series above it.
Data table	The data table provides a detailed breakdown, in tabular format, of the results of a composed aggregation. Tip, a data table is available from many other charts by clicking grey bar at the bottom of the chart.
Line chart	Often the best chart for high density time series. Great for comparing one series to another. Be careful with sparse sets as the connection between points can be misleading.
Markdown widget	Useful for displaying explanations or instructions for dashboards.
Metric	One big number for all of your one big number needs. Perfect for showing a count of hits, or the exact average a numeric field.
Pie chart	Pie charts are ideal for displaying the parts of some whole. For example, sales percentages by department.Pro Tip: Pie charts are best used sparingly, and with no more than 7 slices per pie.
Tile map	Your source for geographic maps. Requires an elasticsearch geo_point field. More specifically, a field that is mapped as type:geo_point with latitude and longitude coordinates.
Vertical bar chart	The goto chart for oh-so-many needs. Great for time and non-time data. Stacked or grouped, exact numbers or percentages. If you are not sure which chart you need, you could do worse than to start here.

图-10

11）选择 from a new serach，如图-11 所示：



Select a search source

From a new search

From a saved search

图-11

12）选择 Spilt Slices，如图-12 所示：

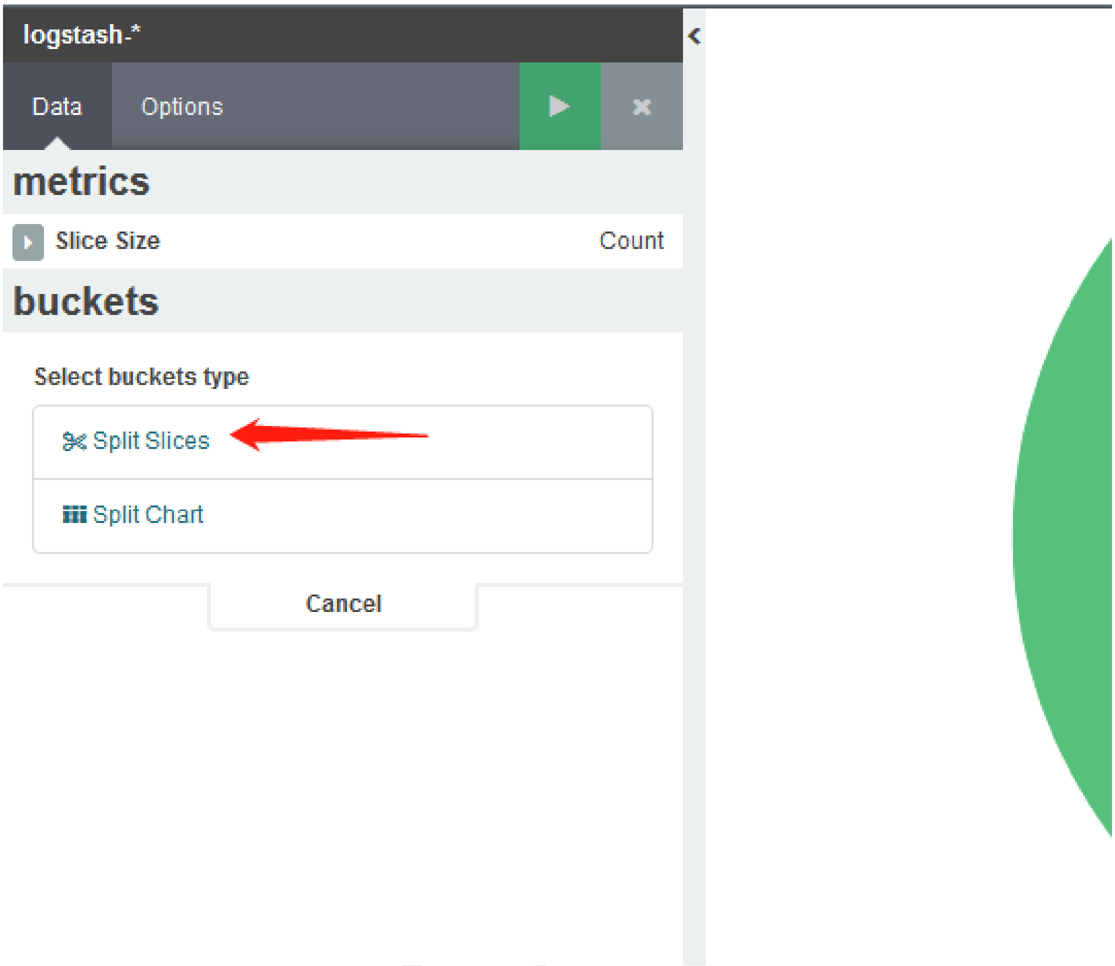
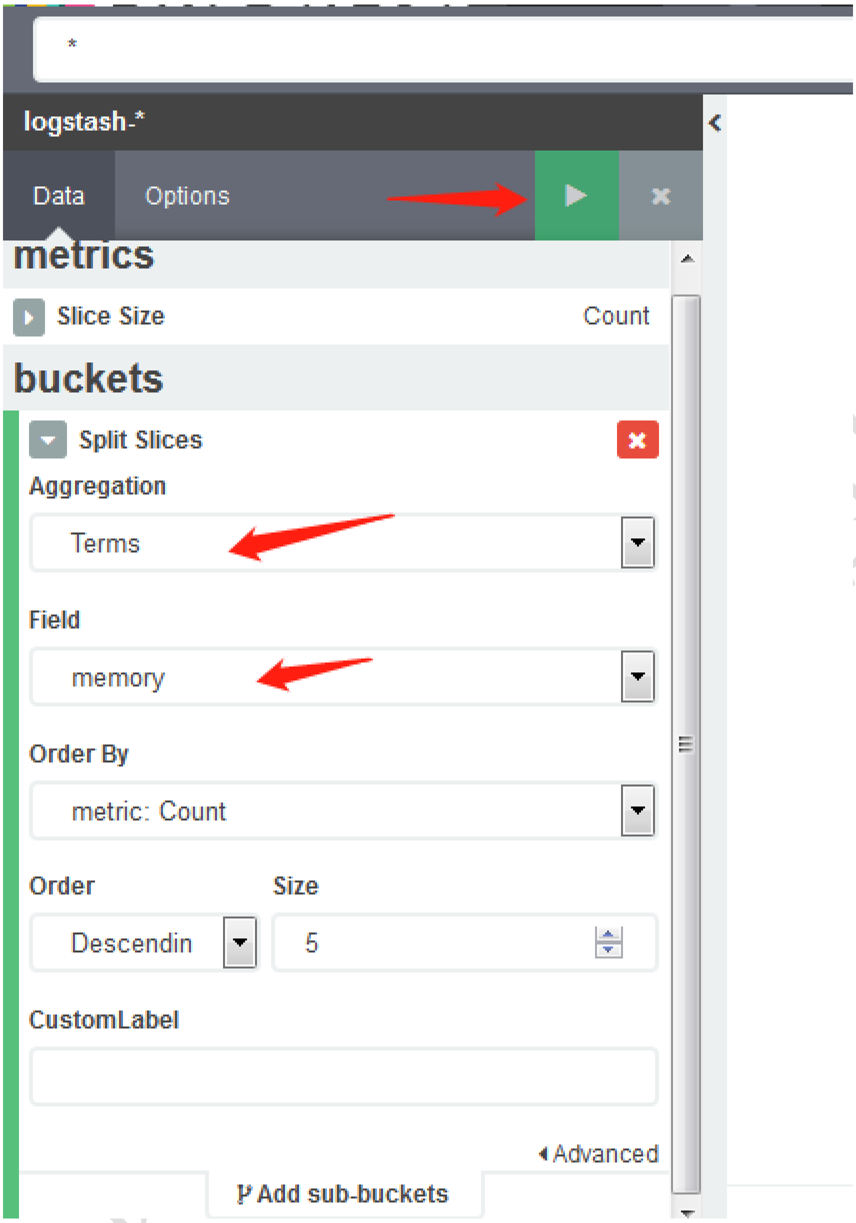


图-12

13）选择 Trems,Memary(也可以选择其他的，这个不固定)，如图-13 所示：



Slice Size Count

Split Slices

Aggregation: Terms

Field: memory

Order By: metric: Count

Order: Descendin

Size: 5

CustomLabel

图-13

14）结果，如图-14 所示：

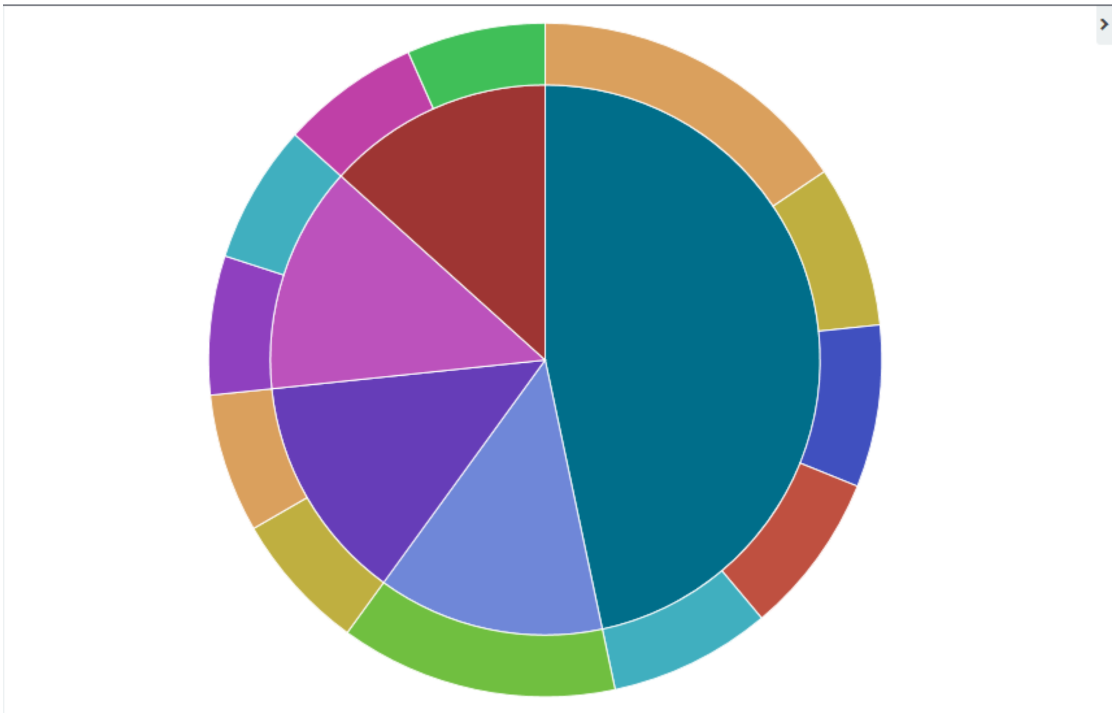


图-14

15）保存后可以在 Dashboard 查看，如图-15 所示：

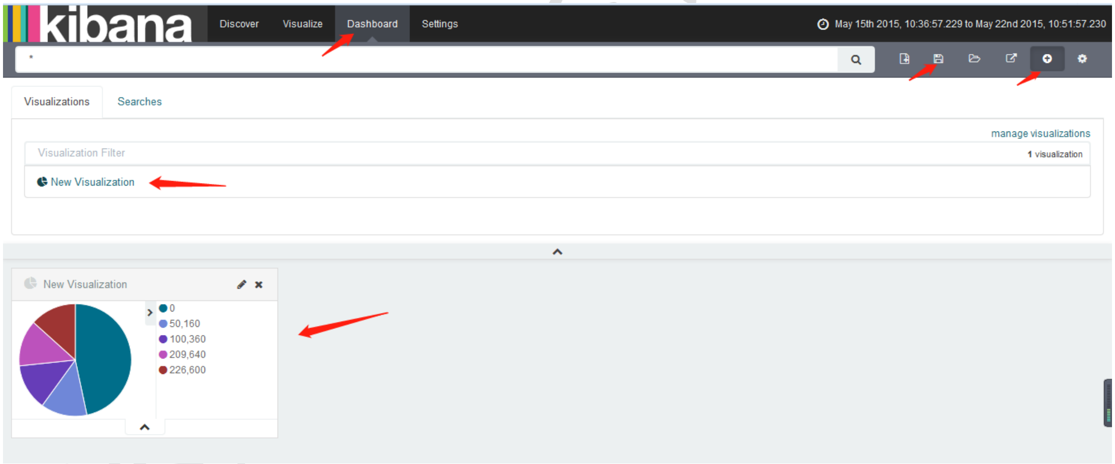


图-15

2. 案例2：综合练习

- **问题**

本案例要求：

- 练习插件

- 安装一台 Apache 服务并配置
- 使用 filebeat 收集 Apache 服务器的日志
- 使用 grok 处理 filebeat 发送过来的日志
- 存入 elasticsearch

- **步骤**

实现此案例需要按照如下步骤进行。

步骤一：安装 logstash

1）配置主机名，ip 和 yum 源，配置/etc/hosts（请把 se1-se5 和 kibana 主机配置和 logstash 一样的/etc/hosts）

```
[root@logstash ~]# vim /etc/hosts
192.168.1.61 se1
192.168.1.62 se2
192.168.1.63 se3
192.168.1.64 se4
192.168.1.65 se5
192.168.1.66 kibana
192.168.1.67 logstash
```

2）安装 java-1.8.0-openjdk 和 logstash

```
[root@logstash ~]#  yum -y install java-1.8.0-openjdk
[root@logstash ~]# yum -y install logstash
[root@logstash ~]#  java -version
openjdk version "1.8.0_131"
OpenJDK Runtime Environment (build 1.8.0_131-b12)
OpenJDK 64-Bit Server VM (build 25.131-b12, mixed mode)
[root@logstash ~]# touch /etc/logstash/logstash.conf
[root@logstash ~]#  /opt/logstash/bin/logstash  --version
logstash 2.3.4
[root@logstash ~]# /opt/logstash/bin/logstash-plugin  list   //查看插件
...
logstash-input-stdin   //标准输入插件
logstash-output-stdout    //标准输出插件
...
[root@logstash ~]# vim /etc/logstash/logstash.conf
input{
    stdin{

   }
}

filter{

}

output{
    stdout{

   }
}

[root@logstash ~]# /opt/logstash/bin/logstash -f /etc/logstash/logstash.conf
//启动并测试
```

```
Settings: Default pipeline workers: 2
Pipeline main started
aa       //logstash 配置从标准输入读取输入源,然后从标准输出输出到屏幕
2018-09-15T06:19:28.724Z logstash aa
```

备注：若不会写配置文件可以找帮助，插件文档的位置：
https://github.com/logstash-plugins

3）codec 类插件

```
[root@logstash ~]# vim /etc/logstash/logstash.conf
input{
    stdin{
    codec => "json"        //输入设置为编码 json
   }
}

filter{

}

output{
    stdout{
    codec => "rubydebug"       //输出设置为 rubydebug
   }
}
[root@logstash ~]# /opt/logstash/bin/logstash -f /etc/logstash/logstash.conf
Settings: Default pipeline workers: 2
Pipeline main started
{"a":1}
{
            "a" => 1,
      "@version" => "1",
    "@timestamp" => "2018-09-15T06:34:14.538Z",
          "host" => "logstash"
}
```

4）file 模块插件

```
[root@logstash ~]# vim /etc/logstash/logstash.conf
input{
  file {
    path          => [ "/tmp/a.log", "/var/tmp/b.log" ]
   sincedb_path   => "/var/lib/logstash/sincedb"     //记录读取文件的位置
   start_position => "beginning"                //配置第一次读取文件从什么地方开始
   type          => "testlog"                   //类型名称
  }
}

filter{

}

output{
    stdout{
    codec => "rubydebug"
}
}

[root@logstash ~]# touch /tmp/a.log
[root@logstash ~]# touch /var/tmp/b.log
```

```
[root@logstash ~]#  /opt/logstash/bin/logstash -f  /etc/logstash/logstash.conf
```

另开一个终端：写入数据

```
[root@logstash ~]#  echo a1 > /tmp/a.log
[root@logstash ~]#  echo b1 > /var/tmp/b.log
```

之前终端查看：

```
 [root@logstash ~]#  /opt/logstash/bin/logstash -f  /etc/logstash/logstash.conf
Settings: Default pipeline workers: 2
Pipeline main started
{
       "message" => "a1",
      "@version" => "1",
    "@timestamp" => "2018-09-15T06:44:30.671Z",
          "path" => "/tmp/a.log",
          "host" => "logstash",
          "type" => "testlog"
}
{
       "message" => "b1",
      "@version" => "1",
    "@timestamp" => "2018-09-15T06:45:04.725Z",
          "path" => "/var/tmp/b.log",
          "host" => "logstash",
          "type" => "testlog"
}
```

5）tcp、udp 模块插件

```
[root@logstash ~]#  vim /etc/logstash/logstash.conf
input{
  file {
    path          => [ "/tmp/a.log", "/var/tmp/b.log" ]
   sincedb_path   => "/var/lib/logstash/sincedb"
   start_position => "beginning"
   type           => "testlog"
  }
  tcp {
     host => "0.0.0.0"
     port => "8888"
     type => "tcplog"
}
   udp {
     host => "0.0.0.0"
     port => "9999"
     type => "udplog"
}
}

filter{

}
output{
    stdout{
    codec => "rubydebug"
}
}
[root@logstash ~]#  /opt/logstash/bin/logstash -f  /etc/logstash/logstash.conf
//启动
```

另开一个终端查看，可以看到端口

```
    [root@logstash tmp]#  netstat -antup | grep 8888
    tcp6        0       0 :::8888                   :::*                    LISTEN
22191/java
    [root@logstash tmp]# netstat -antup | grep 9999
    udp6              0             0  :::9999                                 :::*
22191/java
```

在另一台主机上写一个脚本，发送数据，使启动的 logstash 可以接收到数据

```
    [root@se5 ~]# vim tcp.sh
    function sendmsg(){
      if [[ "$1" == "tcp" ]];then
             exec 9<>/dev/tcp/192.168.1.67/8888
       else
             exec 9<>/dev/udp/192.168.1.67/9999
       fi
          echo "$2" >&9
          exec 9<&-
    }
    [root@se5 ~]# . tcp.sh          //重新载入一下
    [root@se5 ~]# sendmsg udp "is tcp test"
    [root@se5 ~]# sendmsg udp "is tcp ss"
```

logstash 主机查看结果

```
    [root@logstash ~]#  /opt/logstash/bin/logstash -f  /etc/logstash/logstash.conf
    Settings: Default pipeline workers: 2
    Pipeline main started
    {
           "message" => "is tcp test\n",
          "@version" => "1",
        "@timestamp" => "2018-09-15T07:45:00.638Z",
              "type" => "udplog",
              "host" => "192.168.1.65"
    }
    {
           "message" => "is tcp ss\n",
          "@version" => "1",
        "@timestamp" => "2018-09-15T07:45:08.897Z",
              "type" => "udplog",
              "host" => "192.168.1.65"
    }
```

6）syslog 插件练习

```
    [root@logstash ~]#  systemctl  list-unit-files | grep syslog
    rsyslog.service                             enabled
    syslog.socket                               static
    [root@logstash ~]#  vim /etc/logstash/logstash.conf
       start_position => "beginning"
       type          => "testlog"
      }
      tcp {
         host => "0.0.0.0"
         port => "8888"
         type => "tcplog"
    }
       udp {
         host => "0.0.0.0"
         port => "9999"
         type => "udplog"
    }
      syslog {
         port => "514"
         type => "syslog"
      }
    }

    filter{

    }

    output{
        stdout{
        codec => "rubydebug"
    }
    }
```

另一个终端查看是否检测到 514

```
    [root@logstash ~]#  netstat -antup | grep 514
    tcp6        0       0 :::514                    :::*                      LISTEN
22728/java
    udp6              0             0  :::514                                      :::*
22728/java
```

另一台主机上面操作,本地写的日志本地可以查看

```
    [root@se5 ~]# vim /etc/rsyslog.conf
    local0.info                                 /var/log/mylog  //自己添加这一行
    [root@se5 ~]# systemctl restart rsyslog    //重启 rsyslog
    [root@se5 ~]#  ll /var/log/mylog        //提示没有那个文件或目录
    ls: cannot access /var/log/mylog: No such file or directory
    [root@se5 ~]# logger -p local0.info -t nsd "elk"        //写日志
    [root@se5 ~]#  ll /var/log/mylog        //再次查看，有文件
    -rw------- 1 root root 29 Sep 15 16:23 /var/log/mylog
    [root@se5 ~]# tail  /var/log/mylog   //可以查看到写的日志
    Sep 15 16:23:25 se5 nsd: elk
    [root@se5 ~]# tail  /var/log/messages
    //可以查看到写的日志，因为配置文件里有写以.info 结尾的可以收到
    ...
    Sep 15 16:23:25 se5 nsd: elk
```

把本地的日志发送给远程 1.67

```
    [root@se5 ~]# vim /etc/rsyslog.conf
    local0.info                     @192.168.1.67:514
    //写一个@或两个@@都可以，一个@代表 udp，两个@@代表 tcp
    [root@se5 ~]# systemctl restart rsyslog
    [root@se5 ~]# logger  -p local0.info -t nds "001 elk"
    [root@logstash bin]# /opt/logstash/bin/logstash -f  /etc/logstash/logstash.conf
    //检测到写的日志
    {
              "message" => "001 elk",
             "@version" => "1",
           "@timestamp" => "2018-09-05T09:15:47.000Z",
                 "type" => "syslog",
                 "host" => "192.168.1.65",
             "priority" => 134,
            "timestamp" => "Jun  5 17:15:47",
            "logsource" => "kibana",
              "program" => "nds1801",
             "severity" => 6,
             "facility" => 16,
       "facility_label" => "local0",
       "severity_label" => "Informational"
}
```

rsyslog.conf 配置向远程发送数据，远程登陆 1.65 的时候，把登陆日志的信息（/var/log/secure）转发给 logstash 即 1.67 这台机器

```
[root@se5 ~]#  vim /etc/rsyslog.conf
57 authpriv.*                                        @@192.168.1.67:514
//57 行的/var/log/secure 改为@@192.168.1.67:514
[root@se5 ~]# systemctl restart rsyslog
[root@logstash ~]# /opt/logstash/bin/logstash -f  /etc/logstash/logstash.conf
//找一台主机登录 1.65，logstash 主机会有数据
Settings: Default pipeline workers: 2
Pipeline main started
{
              "message" => "Accepted password for root from 192.168.1.254 port 33780 ssh2\n",
             "@version" => "1",
           "@timestamp" => "2018-09-15T08:40:57.000Z",
                 "type" => "syslog",
                 "host" => "192.168.1.65",
             "priority" => 86,
            "timestamp" => "Sep 15 16:40:57",
            "logsource" => "se5",
              "program" => "sshd",
                  "pid" => "26133",
             "severity" => 6,
             "facility" => 10,
       "facility_label" => "security/authorization",
       "severity_label" => "Informational"
}
{
              "message" => "pam_unix(sshd:session): session opened for user root by (uid=0)\n",
             "@version" => "1",
           "@timestamp" => "2018-09-15T08:40:57.000Z",
                 "type" => "syslog",
                 "host" => "192.168.1.65",
             "priority" => 86,
            "timestamp" => "Sep 15 16:40:57",
            "logsource" => "se5",
              "program" => "sshd",
                  "pid" => "26133",
             "severity" => 6,
             "facility" => 10,
       "facility_label" => "security/authorization",
       "severity_label" => "Informational"
```

7）filter grok 插件

grok 插件：

解析各种非结构化的日志数据插件

grok 使用正则表达式把飞结构化的数据结构化

在分组匹配，正则表达式需要根据具体数据结构编写

虽然编写困难，但适用性极广

```
    [root@logstash ~]#  vim /etc/logstash/logstash.conf
    input{
           stdin{ codec => "json" }
      file {
        path          => [ "/tmp/a.log", "/var/tmp/b.log" ]
       sincedb_path   => "/var/lib/logstash/sincedb"
       start_position => "beginning"
       type           => "testlog"
      }
      tcp {
         host => "0.0.0.0"
         port => "8888"
         type => "tcplog"
    }
       udp {
         host => "0.0.0.0"
         port => "9999"
         type => "udplog"
    }
      syslog {
         port => "514"
         type => "syslog"
      }
    }

    filter{
       grok{
            match => ["message", "(?<key>reg)"]
      }
    }

    output{
        stdout{
        codec => "rubydebug"
    }
    }
    [root@se5 ~]# yum -y install httpd
    [root@se5 ~]# systemctl restart httpd
    [root@se5 ~]# vim /var/log/httpd/access_log
    192.168.1.254  -  -  [15/Sep/2018:18:25:46  +0800]  "GET  /  HTTP/1.1"  403  4897  "-"
"Mozilla/5.0 (Windows NT 6.1; WOW64; rv:52.0) Gecko/20100101 Firefox/52.0"
```

复制/var/log/httpd/access_log 的日志到 logstash 下的/tmp/a.log

```
    [root@logstash ~]# vim /tmp/a.log
    192.168.1.254  -  -  [15/Sep/2018:18:25:46  +0800]  "GET  /  HTTP/1.1"  403  4897  "-"
"Mozilla/5.0 (Windows NT 6.1; WOW64; rv:52.0) Gecko/20100101 Firefox/52.0"

    [root@logstash ~]#  /opt/logstash/bin/logstash -f  /etc/logstash/logstash.conf
    //出现 message 的日志，但是没有解析是什么意思
    Settings: Default pipeline workers: 2
    Pipeline main started
    {
          "message" => ".168.1.254 - - [15/Sep/2018:18:25:46 +0800] \"GET / HTTP/1.1\"
403  4897  \"-\"  \"Mozilla/5.0  (Windows  NT  6.1;  WOW64;  rv:52.0)  Gecko/20100101
Firefox/52.0\"",
         "@version" => "1",
       "@timestamp" => "2018-09-15T10:26:51.335Z",
             "path" => "/tmp/a.log",
             "host" => "logstash",
             "type" => "testlog",
             "tags" => [
           [0] "_grokparsefailure"
       ]
    }
```

若要解决没有解析的问题，同样的方法把日志复制到/tmp/a.log，logstash.conf 配置文件里面修改 grok

查找正则宏路径

```
    [root@logstash ~]# cd  /opt/logstash/vendor/bundle/ \
    jruby/1.9/gems/logstash-patterns-core-2.0.5/patterns/
    [root@logstash ~]# vim grok-patterns   //查找 COMBINEDAPACHELOG
    COMBINEDAPACHELOG %{COMMONAPACHELOG} %{QS:referrer} %{QS:agent}

    [root@logstash ~]#  vim /etc/logstash/logstash.conf
    ...
    filter{
       grok{
            match => ["message", "%{COMBINEDAPACHELOG}"]
      }
    }
    ...
```

解析出的结果

```
     [root@logstash ~]#  /opt/logstash/bin/logstash -f  /etc/logstash/logstash.conf
    Settings: Default pipeline workers: 2
    Pipeline main started
    {
            "message"  =>  "192.168.1.254  -  -  [15/Sep/2018:18:25:46  +0800]  \"GET
/noindex/css/open-sans.css HTTP/1.1\" 200 5081 \"http://192.168.1.65/\" \"Mozilla/5.0
(Windows NT 6.1; WOW64; rv:52.0) Gecko/20100101 Firefox/52.0\"",
           "@version" => "1",
         "@timestamp" => "2018-09-15T10:55:57.743Z",
               "path" => "/tmp/a.log",
               "host" => "logstash",
               "type" => "testlog",
           "clientip" => "192.168.1.254",
              "ident" => "-",
               "auth" => "-",
          "timestamp" => "15/Sep/2018:18:25:46 +0800",
               "verb" => "GET",
            "request" => "/noindex/css/open-sans.css",
        "httpversion" => "1.1",
           "response" => "200",
              "bytes" => "5081",
           "referrer" => "\"http://192.168.1.65/\"",
              "agent" => "\"Mozilla/5.0 (Windows NT 6.1; WOW64; rv:52.0) Gecko/20100101
Firefox/52.0\""
    }
```

步骤二： 安装 Apache 服务，用 filebeat 收集 Apache 服务器的日志，存入 elasticsearch

1）在之前安装了 Apache 的主机上面安装 filebeat

```
    [root@se5 ~]#  yum -y install filebeat
    [root@se5 ~]#  vim/etc/filebeat/filebeat.yml
```

```
paths:
 - /var/log/httpd/access_log   //日志的路径，短横线加空格代表 yml 格式
document_type: apachelog    //文档类型
elasticsearch:        //加上注释
hosts: ["localhost:9200"]              //加上注释
logstash:                   //去掉注释
hosts: ["192.168.1.67:5044"]   //去掉注释,logstash 那台主机的 ip
[root@se5 ~]# systemctl start filebeat

[root@logstash ~]#  vim /etc/logstash/logstash.conf
input{
       stdin{ codec => "json" }
       beats{
           port => 5044
}
  file {
    path          => [ "/tmp/a.log", "/var/tmp/b.log" ]
   sincedb_path   => "/dev/null"
   start_position => "beginning"
   type          => "testlog"
  }
  tcp {
     host => "0.0.0.0"
     port => "8888"
     type => "tcplog"
}
   udp {
     host => "0.0.0.0"
     port => "9999"
     type => "udplog"
}
  syslog {
     port => "514"
     type => "syslog"
  }
}

filter{
if [type] == "apachelog"{
   grok{
        match => ["message", "%{COMBINEDAPACHELOG}"]
  }}
}

output{
      stdout{ codec => "rubydebug" }
      if [type] == "filelog"{
      elasticsearch {
          hosts => ["192.168.1.61:9200", "192.168.1.62:9200"]
          index => "filelog"
          flush_size => 2000
          idle_flush_time => 10
      }}
}
 [root@logstash logstash]#  /opt/logstash/bin/logstash  \
-f  /etc/logstash/logstash.conf
```

打开另一终端查看 5044 是否成功启动

```
    [root@logstash ~]#  netstat -antup | grep 5044
    tcp6        0       0 :::5044                   :::*                      LISTEN
23776/java
```

```
[root@se5 ~]#  firefox 192.168.1.65    //ip 为安装 filebeat 的那台机器
```

回到原来的终端，有数据

2）修改 logstash.conf 文件

```
[root@logstash logstash]# vim logstash.conf
...
output{
      stdout{ codec => "rubydebug" }
      if [type] == "apachelog"{
      elasticsearch {
          hosts => ["192.168.1.61:9200", "192.168.1.62:9200"]
          index => "apachelog"
          flush_size => 2000
          idle_flush_time => 10
      }}
}
```

浏览器访问 Elasticsearch，有 apachelog，如图-16 所示：

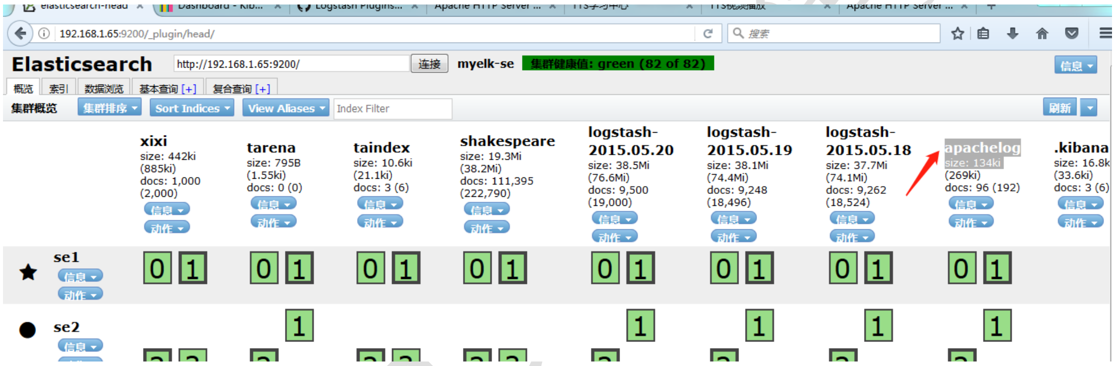

192.168.1.65:9200/_plugin/head/
Elasticsearch
http://192.168.1.65:9200/
连接
myelk-se
集群健康值: green (82 of 82)
信息
概览 索引 数据浏览 基本查询 [+] 复合查询 [+]
集群概览 集群排序 Sort Indices View Aliases Index Filter
刷新
xixi size: 442ki (885ki) docs: 1,000 (2,000)
tarena size: 795B (1.55ki) docs: 0 (0)
taindex size: 10.6ki (21.1ki) docs: 3 (6)
shakespeare size: 19.3Mi (38.2Mi) docs: 111,395 (222,790)
logstash-2015.05.20 size: 38.5Mi (76.6Mi) docs: 9,500 (19,000)
logstash-2015.05.19 size: 38.1Mi (74.4Mi) docs: 9,248 (18,496)
logstash-2015.05.18 size: 37.7Mi (74.1Mi) docs: 9,262 (18,524)
apachelog size: 134ki (269ki) docs: 96 (192)
.kibana size: 16.8k (33.6ki) docs: 3 (6)
信息 动作
se1
se2
0 1

图-16