NSD ARCHITECTURE DAY04

1 案例1：导入数据

1.1 问题

本案例要求批量导入数据：

- 批量导入数据并查看

1.2 步骤

实现此案例需要按照如下步骤进行。

步骤一：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
},
```

Top

```
{
        "_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"
    }
```

Top

```
49.     }, {
50.       "_index" : "xixi",
51.       "_type" : "haha",
52.       "_id" : "25",
53.       "_version" : 1,
54.       "found" : true,
55.       "_source" : {
56.         "account_number" : 25,
57.         "balance" : 40540,
58.         "firstname" : "Virginia",
59.         "lastname" : "Ayala",
60.         "age" : 39,
61.         "gender" : "F",
62.         "address" : "171 Putnam Avenue",
63.         "employer" : "Filodyne",
64.         "email" : "virginiaayala@filodyne.com",
65.         "city" : "Nicholson",
66.         "state" : "PA"
67.       }
68.     } ]
69.   }
```

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：

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

Top

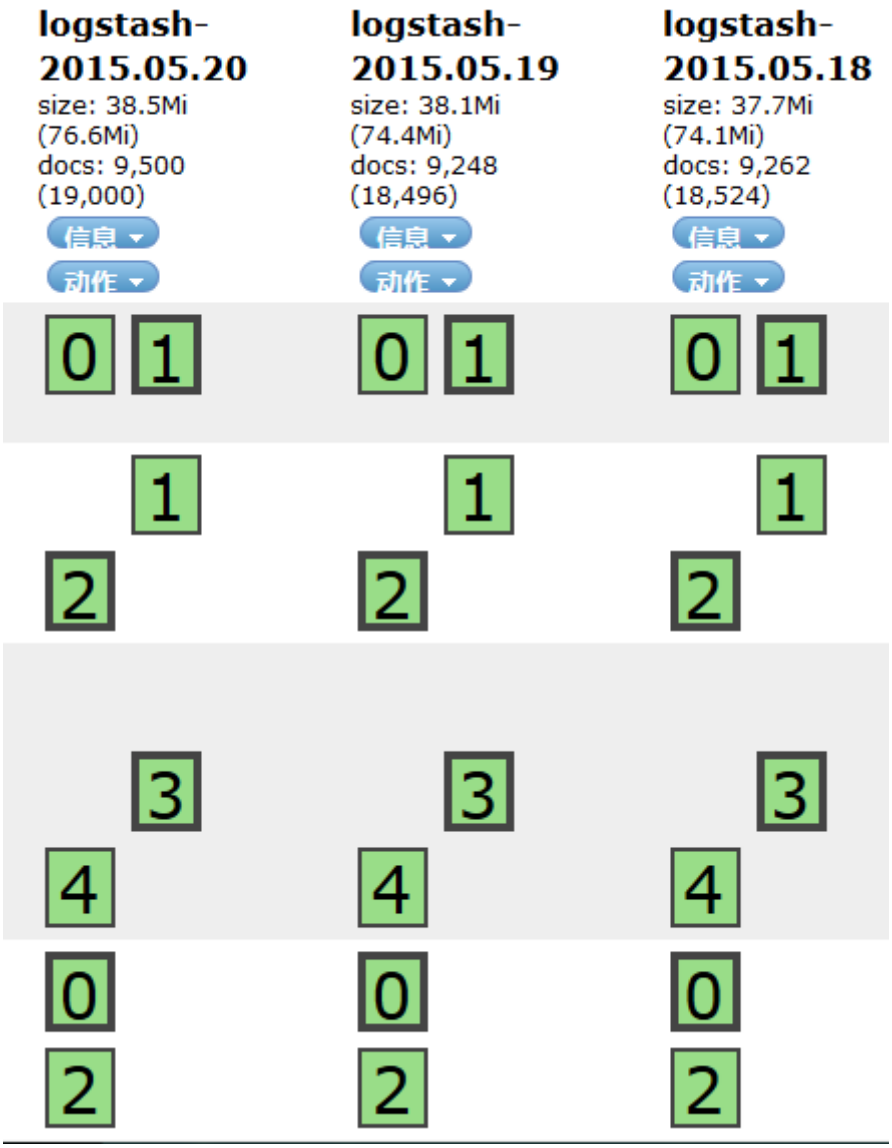
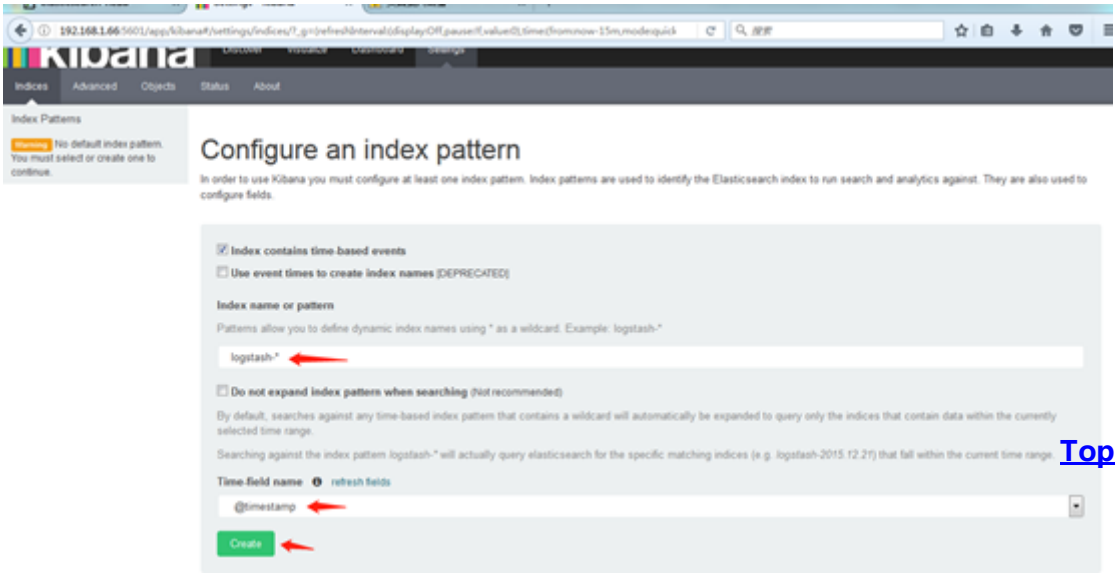


图-1

2）kibana导入数据，如图-2所示：

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



Top

图-2

3）成功创建会有logstash-*，如图-3所示：

/

图-3

4）导入成功之后选择Discover，如图-4所示：

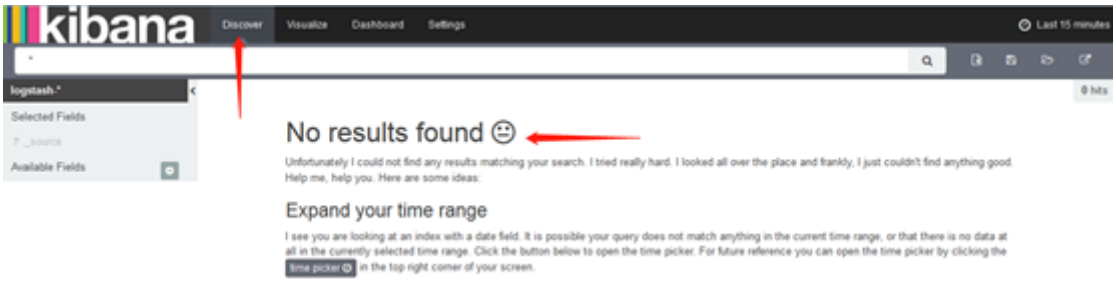


图-4

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：

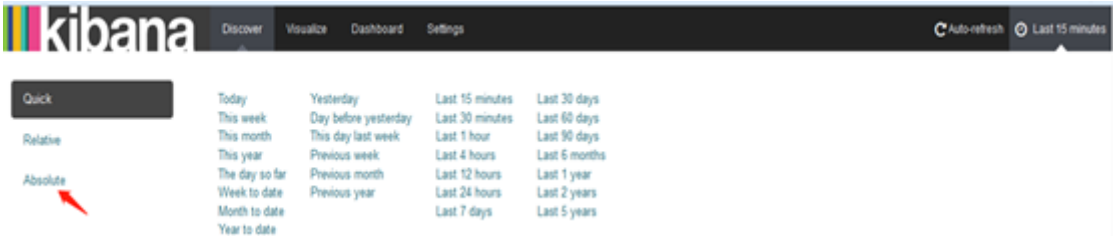


图-6

7）选择时间2015-5-15到2015-5-22，如图-7所示：

Top

图-7

8）查看结果，如图-8所示：

图-8

9）除了柱状图，Kibana还支持很多种展示方式 ，如图-9所示：

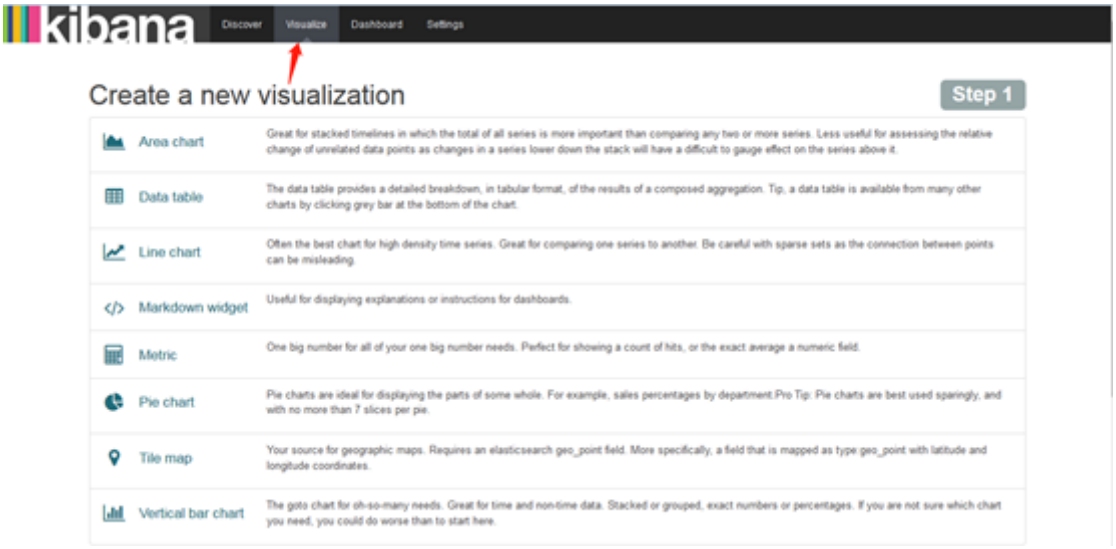


图-9

10）做一个饼图，选择Pie chart，如图-10所示：

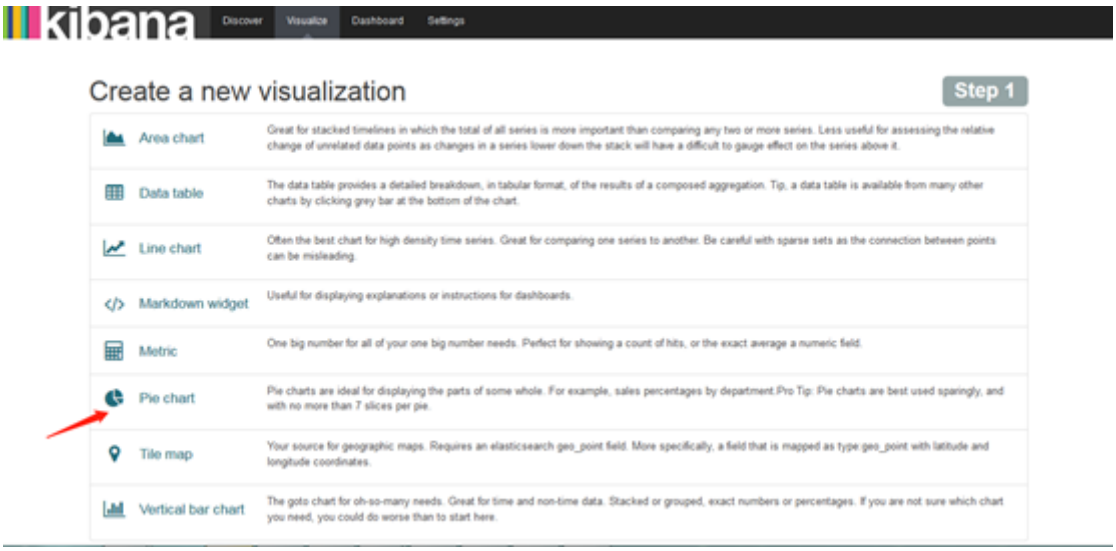
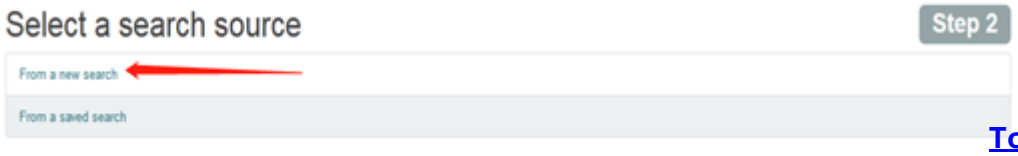


图-10

11）选择from a new serach，如图-11所示：



Top

图-11

12）选择Spilt Slices，如图-12所示：

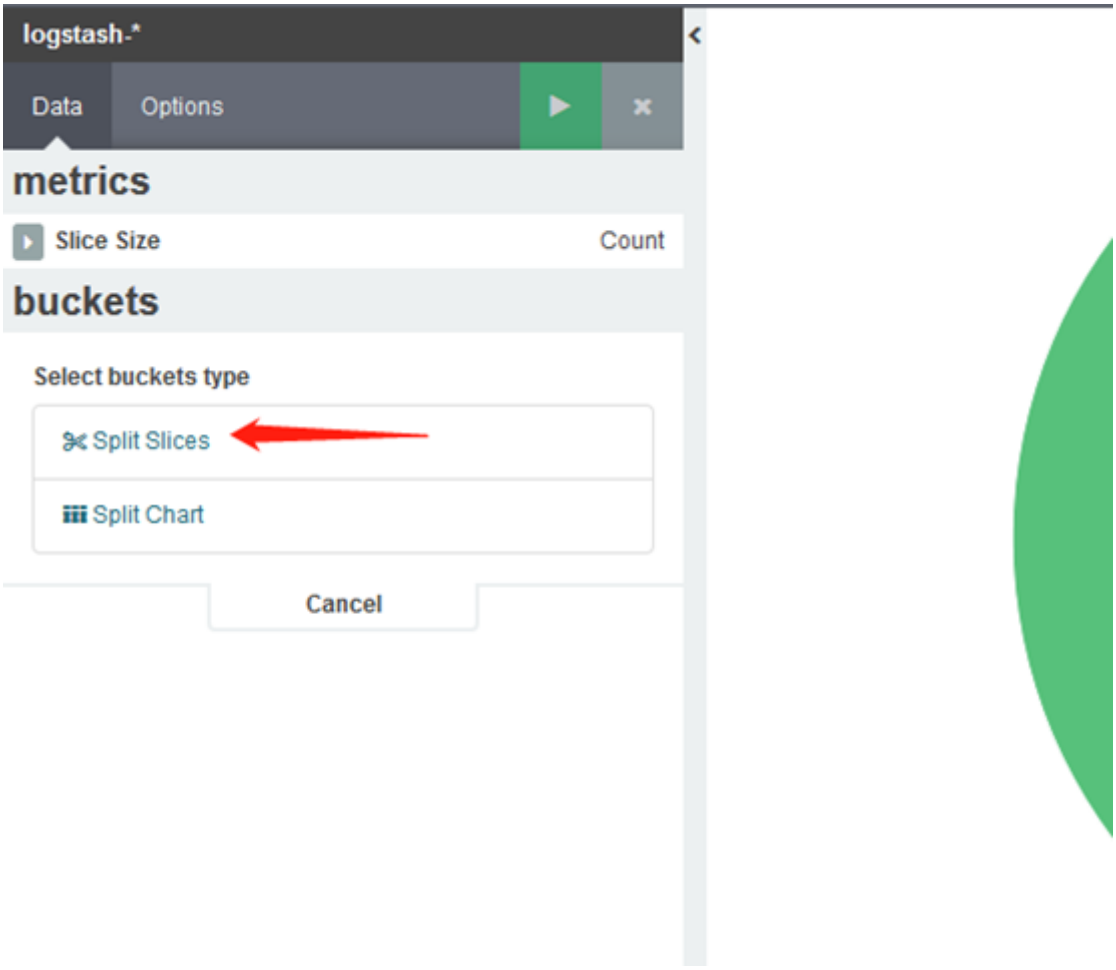


图-12

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

Top

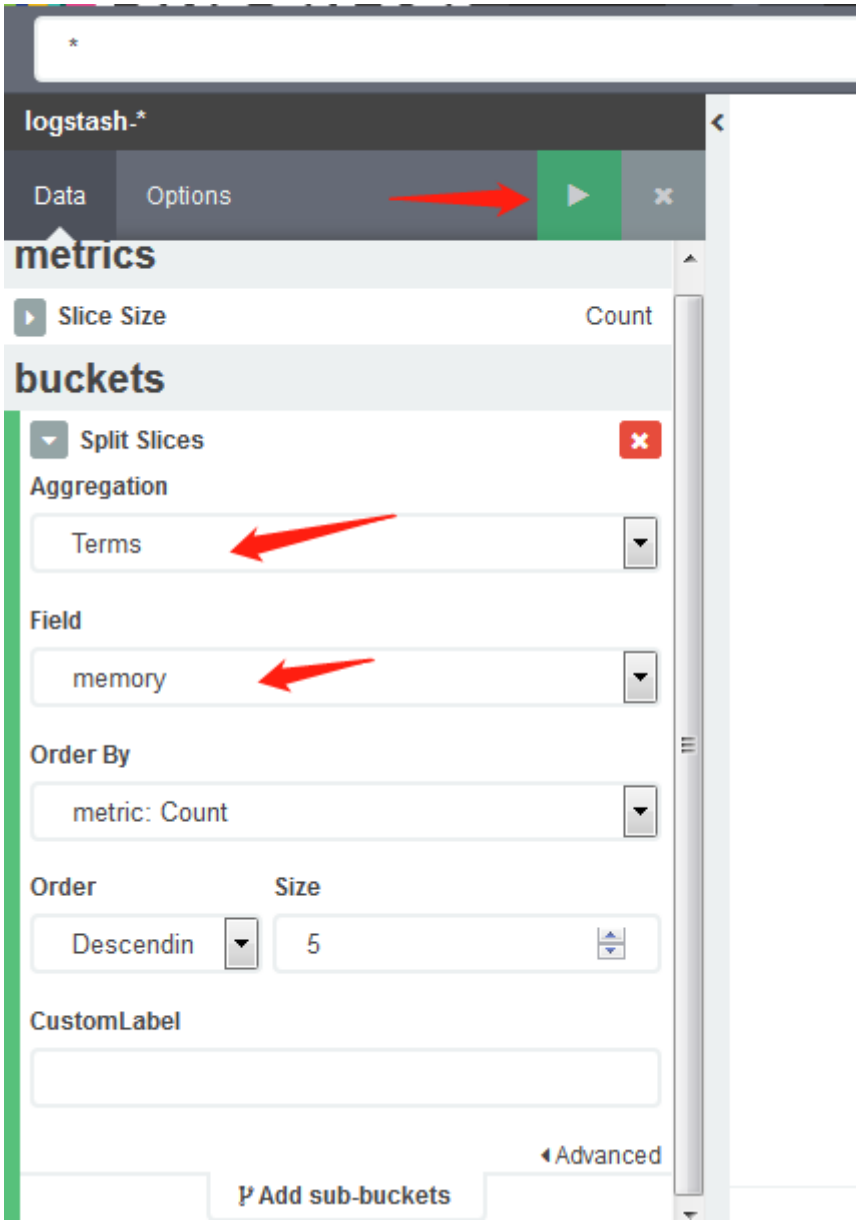
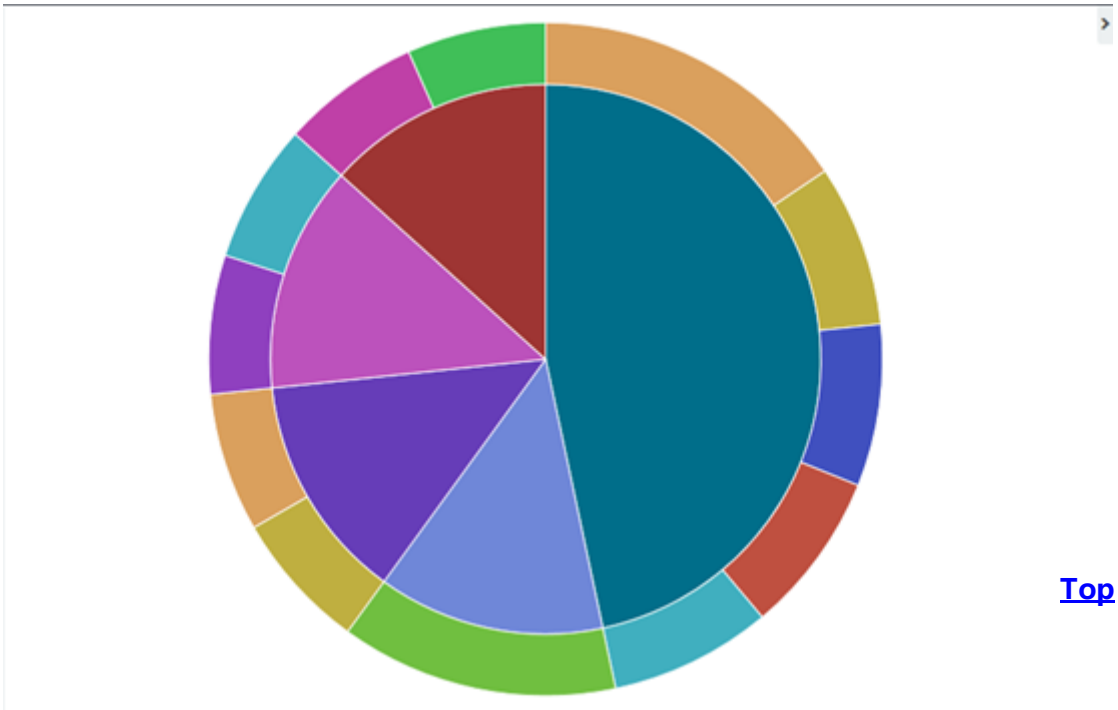


图-13

14）结果，如图-14所示：



Top

图-14

15）保存后可以在Dashboard查看，如图-15所示：

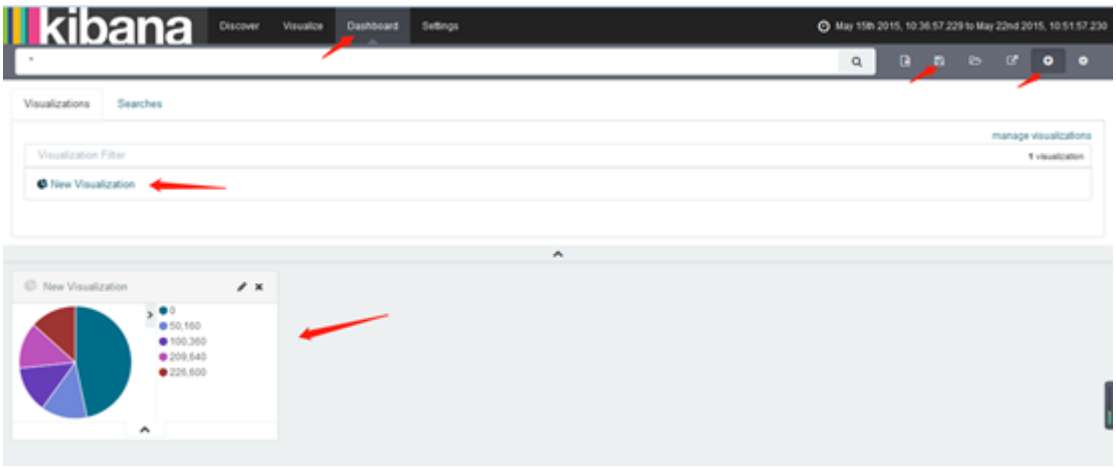


图-15

2 案例2：综合练习

2.1 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

2.2 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
```

Top

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
```

Top

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 => "ruby debug"       //输出设置为ruby debug
    }
}
[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
  }
```

Top

```
09.     }
10.
11.     filter{
12.
13.     }
14.
15.     output{
16.         stdout{
17.         codec => "ruby debug"
18.     }
19.     }
20.
21.     [root@logstash ~]# touch /tmp/a.log
22.     [root@logstash ~]# touch /var/tmp/b.log
23.     [root@logstash ~]#  /opt/logstash/bin/logstash -f  /etc/logstash/logstash.conf
```

另开一个终端：写入数据

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

之前终端查看：

```
01.     [root@logstash ~]#  /opt/logstash/bin/logstash -f  /etc/logstash/logstash.conf
02.     Settings: Default pipeline workers: 2
03.     Pipeline main started
04.     {
05.           "message" => "a1",
06.          "@version" => "1",
07.        "@timestamp" => "2018-09-15T06:44:30.671Z",
08.              "path" => "/tmp/a.log",
09.              "host" => "logstash",
10.              "type" => "testlog"
11.     }
12.     {
13.           "message" => "b1",
14.          "@version" => "1",
15.        "@timestamp" => "2018-09-15T06:45:04.725Z",
16.              "path" => "/var/tmp/b.log",
```

Top

```
17.             "host" => "logstash",
18.             "type" => "testlog"
19.     }
20.
```

5）tcp、udp模块插件

```
01.  [root@logstash ~]#  vim /etc/logstash/logstash.conf
02.  input{
03.    file {
04.      path          => [  "/tmp/a.log", "/var/tmp/b.log" ]
05.      sincedb_path   => "/var/lib/logstash/sincedb"
06.      start_position => "beginning"
07.      type          => "testlog"
08.     }
09.     tcp {
10.       host => "0.0.0.0"
11.       port => "8888"
12.       type => "tcplog"
13.  }
14.     udp {
15.       host => "0.0.0.0"
16.       port => "9999"
17.       type => "udplog"
18.  }
19.  }
20.
21.  filter{
22.
23.  }
24.  output{
25.      stdout{
26.      codec => "ruby debug"
27.  }
28.  }
29.  [root@logstash ~]#  /opt/logstash/bin/logstash -f  /etc/logstash/logstash.conf
30.  //启动
```

Top

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",
```

Top

```
16.             "host" => "192.168.1.65"
17.     }
```

6）syslog插件练习

```
01.     [root@logstash ~]# systemctl list-unit-files | grep syslog
02.     rsyslog.service                             enabled
03.     syslog.socket                               static
04.     [root@logstash ~]# vim /etc/logstash/logstash.conf
05.       start_position => "beginning"
06.       type           => "testlog"
07.      }
08.      tcp {
09.        host => "0.0.0.0"
10.        port => "8888"
11.        type => "tcplog"
12.     }
13.       udp {
14.        host => "0.0.0.0"
15.        port => "9999"
16.        type => "udplog"
17.     }
18.      syslog {
19.        port => "514"
20.        type => "syslog"
21.      }
22.     }
23.
24.     filter{
25.
26.     }
27.
28.     output{
29.        stdout{
30.        codec => "ruby debug"
31.     }
32.     }
```

Top

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",
```

Top

```
16.             "logsource" => "kibana",
17.               "program" => "nds1801",
18.              "severity" => 6,
19.              "facility" => 16,
20.         "facility_label" => "local0",
21.         "severity_label" => "Informational"
22.     }
```

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

```
01.  [root@se5 ~]#  vim /etc/rsyslog.conf
02.  57 authpriv.*                                              @@192.168.1.67:514
03.  //57行的/var/log/secure改为@@192.168.1.67:514
04.  [root@se5 ~]# systemctl restart rsyslog
05.  [root@logstash ~]# /opt/logstash/bin/logstash -f  /etc/logstash/logstash.conf
06.  //找一台主机登录1.65，logstash主机会有数据
07.  Settings: Default pipeline workers: 2
08.  Pipeline main started
09.  {
10.             "message" => "Accepted password for root from 192.168.1.254 port 33780 ssh2\r
11.            "@version" => "1",
12.          "@timestamp" => "2018-09-15T08:40:57.000Z",
13.                "type" => "syslog",
14.                "host" => "192.168.1.65",
15.            "priority" => 86,
16.           "timestamp" => "Sep 15 16:40:57",
17.           "logsource" => "se5",
18.             "program" => "sshd",
19.                 "pid" => "26133",
20.            "severity" => 6,
21.            "facility" => 10,
22.       "facility_label" => "security/authorization",
23.       "severity_label" => "Informational"
24.  }
25.  {
26.             "message" => "pam_unix(sshd:session): session opened for user root by (uid=0) \
27.            "@version" => "1",
28.          "@timestamp" => "2018-09-15T08:40:57.000Z",
29.                "type" => "syslog",
```

Top

```
30.                 "host" => "192.168.1.65",
31.              "priority" => 86,
32.             "timestamp" => "Sep 15 16:40:57",
33.             "logsource" => "se5",
34.               "program" => "sshd",
35.                   "pid" => "26133",
36.              "severity" => 6,
37.              "facility" => 10,
38.         "facility_label" => "security/authorization",
39.         "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

```
01.  [root@logstash ~]# vim /etc/logstash/logstash.conf
02.  input{
03.          stdin{ codec => "json" }
04.    file {
05.      path          => [ "/tmp/a.log", "/var/tmp/b.log" ]
06.      sincedb_path  => "/var/lib/logstash/sincedb"
07.      start_position => "beginning"
08.      type          => "testlog"
09.    }
10.    tcp {
11.      host => "0.0.0.0"
12.      port => "8888"
13.      type => "tcplog"
14.  }
15.    udp {
16.      host => "0.0.0.0"
17.      port => "9999"
18.      type => "udplog"
19.  }
20.    syslog {
21.      port => "514"
```

Top

```
22.         type => "syslog"
23.       }
24.     }
25.
26.     filter{
27.       grok{
28.           match => [ "message", "(?<key>reg)" ]
29.       }
30.     }
31.
32.     output{
33.       stdout{
34.       codec => "ruby debug"
35.     }
36.     }
37.     [root@se5 ~]# yum -y install httpd
38.     [root@se5 ~]# systemctl restart httpd
39.     [root@se5 ~]# vim /var/log/httpd/access_log
40.     192.168.1.254 - - [15/Sep/2018:18:25:46 +0800] "GET / HTTP/1.1" 403 4897 "-" "Mozilla
```

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

```
01.     [root@logstash ~]# vim /tmp/a.log
02.     192.168.1.254 - - [15/Sep/2018:18:25:46 +0800] "GET / HTTP/1.1" 403 4897 "-" "Mozilla
03.
04.     [root@logstash ~]#  /opt/logstash/bin/logstash -f  /etc/logstash/logstash.conf
05.     //出现message的日志，但是没有解析是什么意思
06.     Settings: Default pipeline workers: 2
07.     Pipeline main started
08.     {
09.           "message" => ".168.1.254 - - [15/Sep/2018:18:25:46 +0800] \"GET / HTTP/1.1\" 4(
10.          "@version" => "1",
11.        "@timestamp" => "2018-09-15T10:26:51.335Z",
12.              "path" => "/tmp/a.log",
13.              "host" => "logstash",
14.              "type" => "testlog",
15.              "tags" => [
16.           [0] "_grokparsefailure"
17.         ]
```

Top

```
18.     }
```

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

```
01.  [root@logstash ~]# cd  /opt/logstash/vendor/bundle/ \
02.  jruby/1.9/gems/logstash-patterns-core-2.0.5/patterns/
03.  [root@logstash ~]# vim grok-patterns  //查找COMBINEDAPACHELOG
04.  COMBINEDAPACHELOG %{COMMONAPACHELOG} %{QS:referrer} %{QS:agent}
05.
06.  [root@logstash ~]#  vim /etc/logstash/logstash.conf
07.  ...
08.  filter{
09.    grok{
10.         match => ["message", "%{COMBINEDAPACHELOG}"]
11.    }
12.  }
13.  ...
```

解析出的结果

```
01.   [root@logstash ~]#  /opt/logstash/bin/logstash -f  /etc/logstash/logstash.conf
02.  Settings: Default pipeline workers: 2
03.  Pipeline main started
04.  {
05.         "message" => "192.168.1.254 - - [15/Sep/2018:18:25:46 +0800] \"GET /noindex/cs
06.        "@version" => "1",
07.      "@timestamp" => "2018-09-15T10:55:57.743Z",
08.            "path" => "/tmp/a.log",
09.            "host" => "logstash",
10.            "type" => "testlog",
11.        "clientip" => "192.168.1.254",
12.           "ident" => "-",
13.            "auth" => "-",
14.       "timestamp" => "15/Sep/2018:18:25:46 +0800",
15.            "verb" => "GET",
16.         "request" => "/noindex/css/open-sans.css",
```

Top

```
17.       "httpversion" => "1.1",
18.         "response" => "200",
19.           "bytes" => "5081",
20.         "referrer" => "\"http://192.168.1.65/\"",
21.           "agent" => "\"Mozilla/5.0 (Windows NT 6.1; WOW64; rv:52.0) Gecko/20100101 Fir
22.     }
```

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

```
01.     [root@se5 ~]# yum -y install filebeat
02.     [root@se5 ~]# vim/etc/filebeat/filebeat.yml
03.     paths:
04.       - /var/log/httpd/access_log   //日志的路径，短横线加空格代表yml格式
05.     document_type: apachelog    //文档类型
06.     elasticsearch:        //加上注释
07.     hosts: ["localhost:9200"]              //加上注释
08.     logstash:                  //去掉注释
09.     hosts: ["192.168.1.67:5044"]     //去掉注释,logstash那台主机的ip
10.     [root@se5 ~]# systemctl start filebeat
11.
12.     [root@logstash ~]# vim /etc/logstash/logstash.conf
13.     input{
14.           stdin{ codec => "json" }
15.           beats{
16.              port => 5044
17.     }
18.       file {
19.         path          => [ "/tmp/a.log", "/var/tmp/b.log" ]
20.         sincedb_path   => "/dev/null"
21.         start_position => "beginning"
22.         type           => "testlog"
23.       }
24.       tcp {
25.         host => "0.0.0.0"
26.         port => "8888"
27.         type => "tcplog"
28.     }
29.       udp {
```

Top

```
30.         host => "0.0.0.0"
31.         port => "9999"
32.         type => "udplog"
33.     }
34.      syslog {
35.         port => "514"
36.         type => "syslog"
37.      }
38.     }
39.
40.     filter{
41.     if [type] == "apachelog"{
42.       grok{
43.           match => [ "message", "%{COMBINEDAPACHELOG}"]
44.      }}
45.     }
46.
47.     output{
48.          stdout{ codec => "ruby debug" }
49.          if [type] == "filelog"{
50.          elasticsearch {
51.              hosts => [ "192.168.1.61:9200", "192.168.1.62:9200"]
52.              index => "filelog"
53.              flush_size => 2000
54.              idle_flush_time => 10
55.          }}
56.     }
57.      [root@logstash logstash]# /opt/logstash/bin/logstash \
58.     -f /etc/logstash/logstash.conf
```

打开另一终端查看5044是否成功启动

```
01.     [root@logstash ~]# netstat -antup | grep 5044
02.     tcp6       0      0 :::5044                 :::*                    LISTEN      23776/java
03.
04.     [root@se5 ~]# firefox 192.168.1.65  //ip为安装filebeat的那台机器
```

Top

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 => "ruby debug" }
    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：

图-16

Top