[Top](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201904/ARCHITECTURE/DAY04/CASE/01/index.html" \l "page_top_case)

# NSD ARCHITECTURE DAY04

1. [案例1：安装Kibana](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201904/ARCHITECTURE/DAY04/CASE/01/index.html" \l "case1)
2. [案例2：综合练习](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201904/ARCHITECTURE/DAY04/CASE/01/index.html" \l "case2)

## 1 案例1：安装Kibana

### 1.1 问题

本案例要求：

* 安装Kibana
* 配置启动服务查看5601端口是否正常
* 通过web页面访问Kibana

### 1.2 步骤

实现此案例需要按照如下步骤进行

步骤一：安装kibana

1）在另一台主机，配置ip为192.168.1.56，配置yum源，更改主机名

2）安装kibana

1. [root@kibana ~]# yum -y install kibana
2. [root@kibana ~]# rpm -qc kibana
3. /opt/kibana/config/kibana.yml
4. [root@kibana ~]# vim /opt/kibana/config/kibana.yml
5. 2 server.port: 5601
6. //若把端口改为80，可以成功启动kibana，但ss时没有端口，没有监听80端口，服务里面写死了，不能用80端口，只能是5601这个端口
7. 5 server.host: "0.0.0.0"        //服务器监听地址
8. 15 elasticsearch.url: http://192.168.1.51:9200
9. //声明地址，从哪里查，集群里面随便选一个
10. 23 kibana.index: ".kibana"    //kibana自己创建的索引
11. 26 kibana.defaultAppId: "discover"    //打开kibana页面时，默认打开的页面discover
12. 53 elasticsearch.pingTimeout: 1500    //ping检测超时时间
13. 57 elasticsearch.requestTimeout: 30000    //请求超时
14. 64 elasticsearch.startupTimeout: 5000    //启动超时
15. [root@kibana ~]# systemctl restart kibana
16. [root@kibana ~]# systemctl enable kibana
17. Created symlink from /etc/systemd/system/multi-user.target.wants/kibana.service to /usr/lib/systemd/system/kibana.service.
18. [root@kibana ~]# ss -antup | grep 5601 //查看监听端口

3）浏览器访问kibana，如图-1所示：

1. [student@room9pc01 ~]$ firefox 192.168.1.56:5601

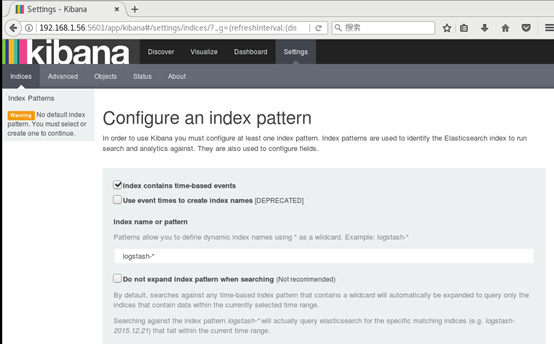


图-1

4）点击Status，查看是否安装成功，全部是绿色的对钩,说明安装成功，如图-2所示：

IMG_257

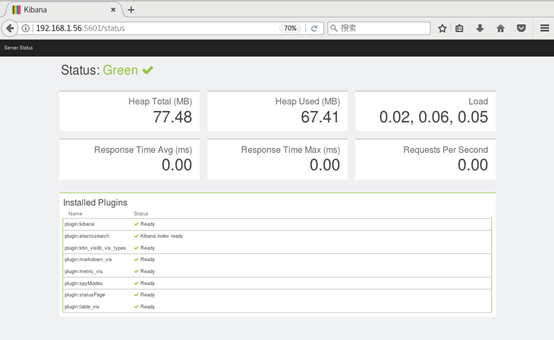


图-2

5）用head插件访问会有.kibana的索引信息，如图-3所示：

1. [student@room9pc01 ~]$ firefox http://192.168.1.55:9200/\_plugin/head

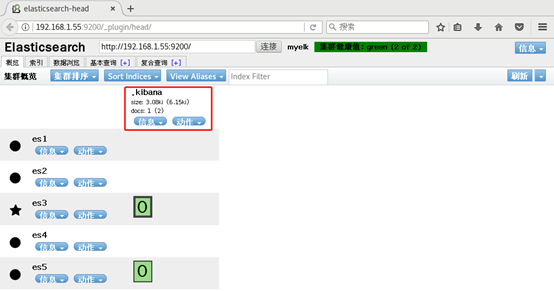


图-3

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-4所示：

1. [student@room9pc01 ~]$ firefox http://192.168.1.55:9200/\_plugin/head

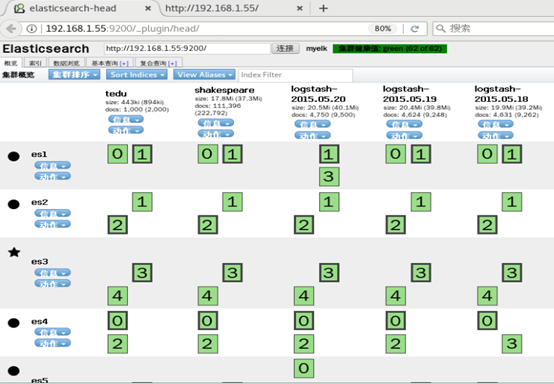


图-4

2）kibana导入数据，如图-5所示：

1. [student@room9pc01 ~]$ firefox http://192.168.1.56:5601

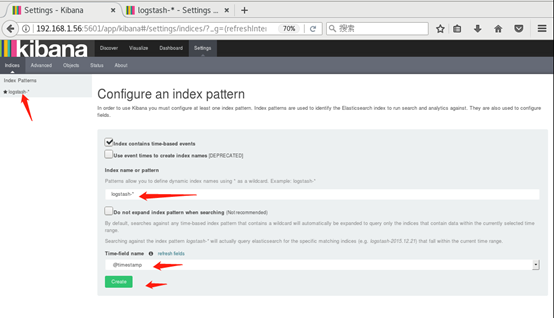


图-5

3）成功创建会有logstash-\*，如图-6所示：



图-6

4）导入成功之后选择Discover，如图-7所示：

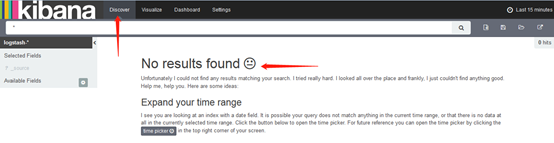


图-7

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-8所示：

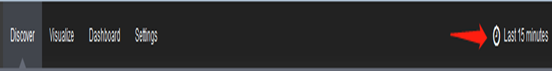


图-8

6）选择Absolute，如图-9所示：

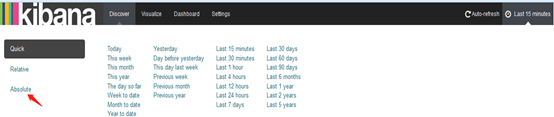


图-9

7）选择时间2015-5-15到2015-5-22，如图-10所示：

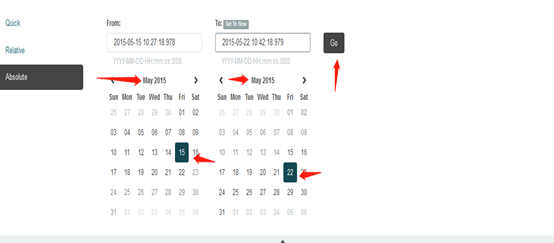


图-10

8）查看结果，如图-11所示：



图-11

9）除了柱状图，Kibana还支持很多种展示方式 ，如图-12所示：

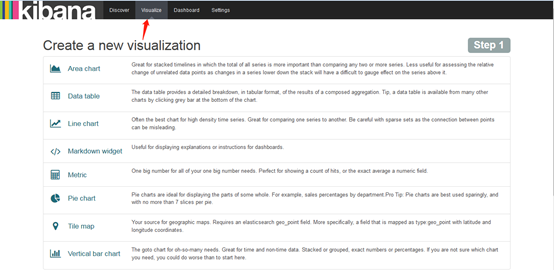


图-12

10）做一个饼图，选择Pie chart，如图-13所示：

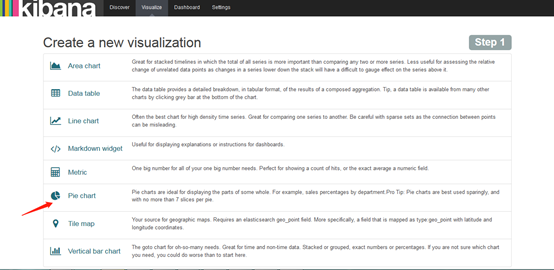


图-14

11）选择from a new serach，如图-11所示：

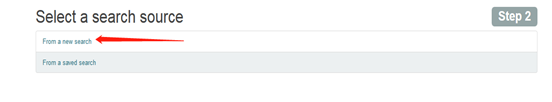


图-15

12）选择Spilt Slices，如图-16所示：

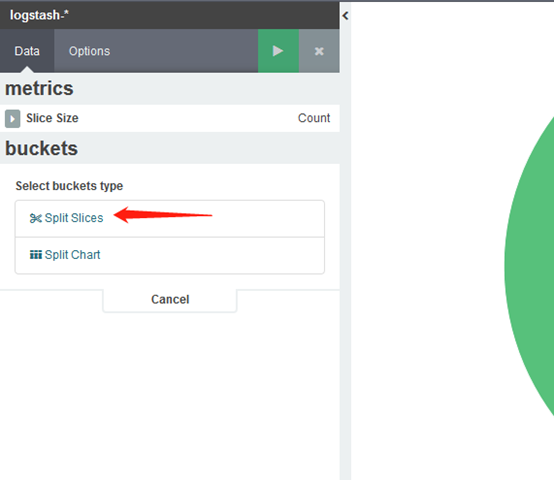


图-16

13）选择Trems,Memary(也可以选择其他的，这个不固定)，如图-17所示：

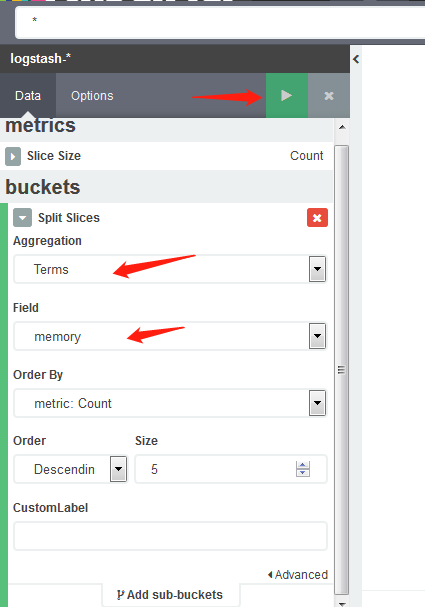


图-17

14）结果，如图-18所示：

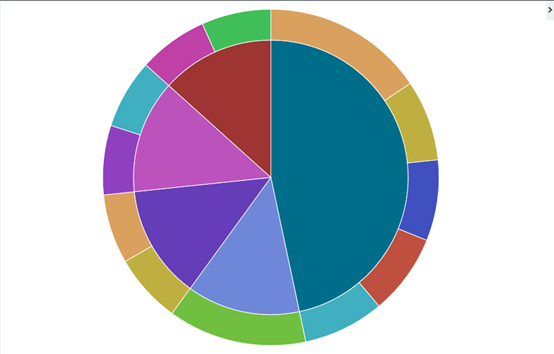


图-18

15）保存后可以在Dashboard查看，如图-19所示：

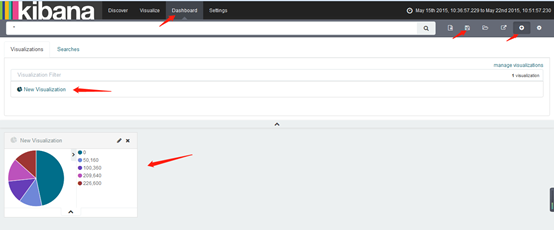


图-19

## 2 案例2：综合练习

### 2.1 问题

本案例要求：

* 安装配置 beats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
* 使用 kibana 做图形展示

### 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es1-es5、kibana主机配置和logstash一样的/etc/hosts）

1. [root@logstash ~]# vim /etc/hosts
2. 192.168.1.51 es1
3. 192.168.1.52 es2
4. 192.168.1.53 es3
5. 192.168.1.54 es4
6. 192.168.1.55 es5
7. 192.168.1.56 kibana
8. 192.168.1.57 logstash

2）安装java-1.8.0-openjdk和logstash

1. [root@logstash ~]# yum -y install java-1.8.0-openjdk
2. [root@logstash ~]# yum -y install logstash
3. [root@logstash ~]# java -version
4. openjdk version "1.8.0\_161"
5. OpenJDK Runtime Environment (build 1.8.0\_161-b14)
6. OpenJDK 64-Bit Server VM (build 25.161-b14, mixed mode)
7. [root@logstash ~]# touch /etc/logstash/logstash.conf
8. [root@logstash ~]# /opt/logstash/bin/logstash --version
9. logstash 2.3.4
10. [root@logstash ~]# /opt/logstash/bin/logstash-plugin list //查看插件
11. ...
12. logstash-input-stdin    //标准输入插件
13. logstash-output-stdout    //标准输出插件
14. ...
15. [root@logstash ~]# vim /etc/logstash/logstash.conf
16. input{
17. stdin{
18. }
19. }
20. filter{
21. }
22. output{
23. stdout{
24. }
25. }
26. [root@logstash ~]# /opt/logstash/bin/logstash -f /etc/logstash/logstash.conf
27. //启动并测试
28. Settings: Default pipeline workers: 2
29. Pipeline main started
30. 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
31. 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

1. [root@logstash ~]# vim /etc/logstash/logstash.conf
2. input{
3. stdin{
4. codec => "json"        //输入设置为编码json
5. }
6. }
7. filter{
8. }
9. output{
10. stdout{
11. codec => "rubydebug"        //输出设置为rubydebug
12. }
13. }
14. [root@logstash ~]# /opt/logstash/bin/logstash -f /etc/logstash/logstash.conf
15. Settings: Default pipeline workers: 2
16. Pipeline main started
17. {"a":1}
18. {
19. "a" => 1,
20. "@version" => "1",
21. "@timestamp" => "2019-03-12T03:25:58.778Z",
22. "host" => "logstash"
23. }

4）file模块插件

1. [root@logstash ~]# vim /etc/logstash/logstash.conf
2. input{
3. file {
4. path => [ "/tmp/a.log", "/tmp/b.log" ]
5. sincedb\_path => "/var/lib/logstash/sincedb"    //记录读取文件的位置
6. start\_position => "beginning"                //配置第一次读取文件从什么地方开始
7. type => "testlog"                    //类型名称
8. }
9. }
10. filter{
11. }
12. output{
13. stdout{
14. codec => "rubydebug"
15. }
16. }
17. [root@logstash ~]# touch /tmp/a.log
18. [root@logstash ~]# touch /tmp/b.log
19. [root@logstash ~]# /opt/logstash/bin/logstash -f /etc/logstash/logstash.conf

另开一个终端：写入数据

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

之前终端查看：

1. [root@logstash ~]# /opt/logstash/bin/logstash -f /etc/logstash/logstash.conf
2. Settings: Default pipeline workers: 2
3. Pipeline main started
4. {
5. "message" => "a1",
6. "@version" => "1",
7. "@timestamp" => "2019-03-12T03:40:24.111Z",
8. "path" => "/tmp/a.log",
9. "host" => "logstash",
10. "type" => "testlog"
11. }
12. {
13. "message" => "b1",
14. "@version" => "1",
15. "@timestamp" => "2019-03-12T03:40:49.167Z",
16. "path" => "/tmp/b.log",
17. "host" => "logstash",
18. "type" => "testlog"
19. }

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

解析Apache的日志

1. [root@es5 ~]# yum -y install httpd
2. [root@es5 ~]# systemctl restart httpd

浏览器访问网页，在/var/log/httpd/access\_log有日志出现

1. [root@es5 ~]# cat /var/log/httpd/access\_log
2. 192.168.1.254 - - [12/Mar/2019:11:51:31 +0800] "GET /favicon.ico HTTP/1.1" 404 209 "-" "Mozilla/5.0 (X11; Linux x86\_64; rv:52.0) Gecko/20100101 Firefox/52.0"
3. [root@logstash ~]# vim /etc/logstash/logstash.conf
4. input{
5. file {
6. path => [ "/tmp/a.log", "/tmp/b.log" ]
7. sincedb\_path => "/var/lib/logstash/sincedb"
8. start\_position => "beginning"
9. type => "testlog"
10. }
11. }
12. filter{
13. grok{
14. match => [ "message", "(?<key>reg)" ]
15. }
16. }
17. output{
18. stdout{
19. codec => "rubydebug"
20. }
21. }

复制/var/log/httpd/access\_log的日志到logstash下的/tmp/a.log

1. [root@logstash ~]# vim /tmp/a.log
2. 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"
3. [root@logstash ~]# /opt/logstash/bin/logstash -f /etc/logstash/logstash.conf
4. //出现message的日志，但是没有解析是什么意思
5. Settings: Default pipeline workers: 2
6. Pipeline main started
7. {
8. "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\"",
9. "@version" => "1",
10. "@timestamp" => "2018-09-15T10:26:51.335Z",
11. "path" => "/tmp/a.log",
12. "host" => "logstash",
13. "type" => "testlog",
14. "tags" => [
15. [0] "\_grokparsefailure"
16. ]
17. }

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

1. [root@logstash ~]# cd /opt/logstash/vendor/bundle/ \
2. jruby/1.9/gems/logstash-patterns-core-2.0.5/patterns/
3. [root@logstash ~]# vim grok-patterns //查找COMBINEDAPACHELOG
4. COMBINEDAPACHELOG %{COMMONAPACHELOG} %{QS:referrer} %{QS:agent}
5. [root@logstash ~]# vim /etc/logstash/logstash.conf
6. ...
7. filter{
8. grok{
9. match => ["message", "%{COMBINEDAPACHELOG}"]
10. }
11. }
12. ...

解析出的结果

1. [root@logstash ~]# /opt/logstash/bin/logstash -f /etc/logstash/logstash.conf
2. Settings: Default pipeline workers: 2
3. Pipeline main started
4. {
5. "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\"",
6. "@version" => "1",
7. "@timestamp" => "2018-09-15T10:55:57.743Z",
8. "path" => "/tmp/a.log",
9. ZZ "host" => "logstash",
10. "type" => "testlog",
11. "clientip" => "192.168.1.254",
12. "ident" => "-",
13. "auth" => "-",
14. "timestamp" => "15/Sep/2019:18:25:46 +0800",
15. "verb" => "GET",
16. "request" => "/noindex/css/open-sans.css",
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 Firefox/52.0\""
22. }
23. ...

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

1）在之前安装了Apache的主机上面安装filebeat

1. [root@se5 ~]# yum -y install filebeat
2. [root@se5 ~]# vim/etc/filebeat/filebeat.yml
3. paths:
4. - /var/log/httpd/access\_log //日志的路径，短横线加空格代表yml格式
5. document\_type: apachelog //文档类型
6. elasticsearch:        //加上注释
7. hosts: ["localhost:9200"]                //加上注释
8. logstash:                    //去掉注释
9. hosts: ["192.168.1.57:5044"]     //去掉注释,logstash那台主机的ip
10. [root@se5 ~]# systemctl start filebeat
11. [root@logstash ~]# vim /etc/logstash/logstash.conf
12. input{
13. stdin{ codec => "json" }
14. beats{
15. port => 5044
16. }
17. file {
18. path => [ "/tmp/a.log", "/tmp/b.log" ]
19. sincedb\_path => "/var/lib/logstash/sincedb"
20. start\_position => "beginning"
21. type => "testlog"
22. }
23. filter{
24. if [type] == "apachelog"{
25. grok{
26. match => ["message", "%{COMBINEDAPACHELOG}"]
27. }}
28. }
29. output{
30. stdout{ codec => "rubydebug" }
31. if [type] == "filelog"{
32. elasticsearch {
33. hosts => ["192.168.1.51:9200", "192.168.1.52:9200"]
34. index => "filelog"
35. flush\_size => 2000
36. idle\_flush\_time => 10
37. }}
38. }
39. [root@logstash logstash]# /opt/logstash/bin/logstash \
40. -f /etc/logstash/logstash.conf

打开另一终端查看5044是否成功启动

1. [root@logstash ~]# netstat -antup | grep 5044
2. tcp6 0 0 :::5044 :::\* LISTEN 23776/java
3. [root@se5 ~]# firefox 192.168.1.55 //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

1. [root@logstash logstash]# vim logstash.conf
2. ...
3. output{
4. stdout{ codec => "rubydebug" }
5. if [type] == "apachelog"{
6. elasticsearch {
7. hosts => ["192.168.1.51:9200", "192.168.1.52:9200"]
8. index => "apachelog"
9. flush\_size => 2000
10. idle\_flush\_time => 10
11. }}
12. }

浏览器访问Elasticsearch，有apachelog，如图-20所示：

1. [student@room9pc01 ~]$ firefox http://192.168.1.55:9200/\_plugin/head

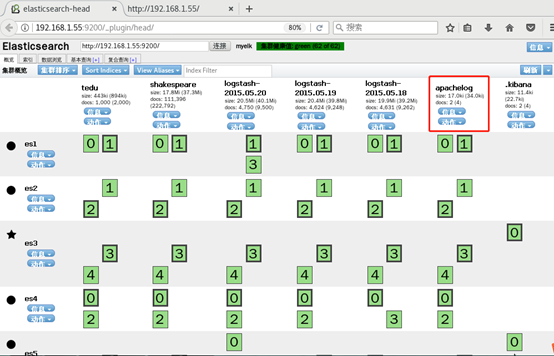


图-20