ELK 集群

1. ELK介绍

1. ELK 不是一款软件，而是一整套解决方案，是 三个软件产品的首字母缩写

– Elasticsearch：负责日志检索和储存

– Logstash：负责日志的收集和分析、处理

– Kibana：负责日志的可视化

2. ELK组件在海量日志系统的运维中，可用于解决：

– 分布式日志数据集中式查询和管理

– 系统监控，包含系统硬件和应用各个组件的监控

– 故障排查

– 安全信息和事件管理

– 报表功能

3.分布式集群的优点

- IO提高了

- 存储空间变大

2. 环境准备

1. 准备6台centos7 x86\_64为的服务器，配置主机名和ip地址，并写入/etc/hosts配置文件中

192.168.1.110 es01

192.168.1.112 es02

192.168.1.113 es03

192.168.1.114 es04

192.168.1.115 es05

192.168.1.116 es06

2. 搭建网络yum源

192.168.1.254 /var/ftp/elk/放入elk需要的所有软件包，并生成yum源

vim /etc/yum.repos.d/dvd.repo [6台]

[dvd]

name=centos7

baseurl=ftp://192.168.1.254/CentOS7

enabled=1

gpgcheck=1

[dvd-elk]

name=centos7-elk

baseurl=ftp://192.168.1.254/elk

enabled=1

gpgcheck=0

3. es0[1-6]安装java-1.8.0-openjdk

yum –y install java-1.8.0-openjdk

ELK之一：elasticsearch 安装使用

1. es0[1-5]安装elasticsearch ——类似mysql数据库

2. 修改elasticsearch配置文件

vim /etc/elasticsearch/elasticsearch.yml

cluster.name: nsd1805

node.name: 当前主机名

network.host: 0.0.0.0

discovery.zen.minimum\_master\_nodes: [“es01”, “es02”, “es03”]

nodes处写几个代表就行，启动的时候先启动这几台

3. 启动elasticsearch服务

systemctl start elasticsearch

监听 9200 和9300端口

4. 查看集群状态

curl <http://192.168.1.115:9200/_cluster/health?pretty>

{

"cluster\_name" : "nsd1805",

"status" : "green",

"timed\_out" : false,

"number\_of\_nodes" : 5,

"number\_of\_data\_nodes" : 5,

"active\_primary\_shards" : 26,

"active\_shards" : 52,

"relocating\_shards" : 0,

"initializing\_shards" : 0,

"unassigned\_shards" : 0,

"delayed\_unassigned\_shards" : 0,

"number\_of\_pending\_tasks" : 0,

"number\_of\_in\_flight\_fetch" : 0,

"task\_max\_waiting\_in\_queue\_millis" : 0,

"active\_shards\_percent\_as\_number" : 100.0

}

查看集群正常，切nodes为5表示elasticsearch 搭建成功

5. 安装ES插件并使用

1. head插件

展现拓扑结构和状态，进行索引和节点操作

2. kopf 插件

elasticsearch 管理工具

3. bigdesk 插件

集群监控工具，查看集群资源使用、内存等信息

4.在192.168.1.115上下载三个插件，浏览器访问插件信息时地址也是这个

/usr/share/elasticsearch/bin/plugin install <ftp://192.168.1.254/elk/elasticsearch-head-master.zip>

/usr/share/elasticsearch/bin/plugin install <ftp://192.168.1.254/elk/elasticsearch-kopf-master.zip>

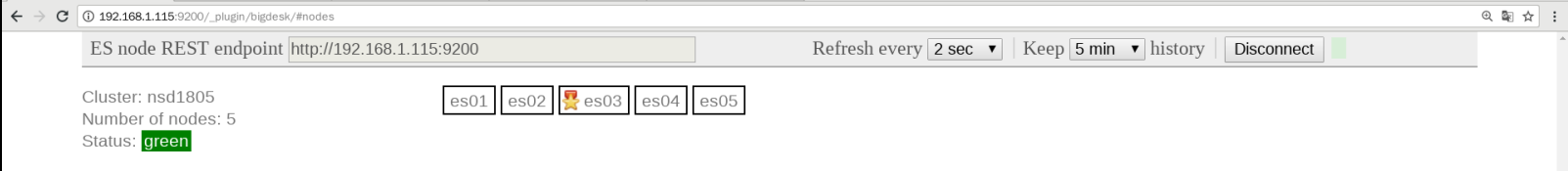
/usr/share/elasticsearch/bin/plugin install <file:///tmp/bigdesk-master.zip>

/usr/share/elasticsearch/bin/plugin list

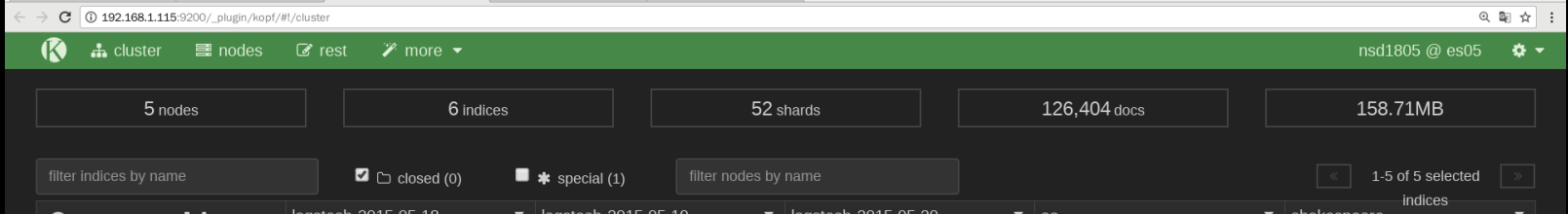
<http://192.168.1.115:9200/_plugin/head>

<http://192.168.1.115:9200/_plugin/kopf>

<http://192.168.1.115:9200/_plugin/bigdesk>







6. elasticsearch 中数据的增删改查

1. 创建索引

curl -XPUT 'http://192.168.1.113:9200/tarena/' -d '{

"settings":{

"index":{

"number\_of\_shards": 5,

"number\_of\_replicas":1

}

}

}'

总结：创建索引的两种方式：

1.head插件页面创建索引

2.命令行PUT方式创建

2. 添加数据

curl -XPUT 'http://192.168.1.113:9200/tarena/1' -d '{

"title": "阶段1",

"name": {"first": "小逗比", "last": "牛犇"},

"age": 25

}'

3. 修改数据

curl -XPUT 'http://192.168.1.113:9200/tarena/1/\_update' -d '{

“doc”: {

“age”: 18

}

}'

4. 查询数据

curl –XGET “<http://192.168.1.113:9200/tarena/1?pretty>”

curl –Xput <http://192.168.1.113:9200/_mget> –d “{

docs: [

{

“\_index”: “tarena”,

“\_type”: “nsd”,

“\_id”: 20

},

… …

]

}”

5. 删除数据

curl –XDELETE <http://192.168.1.115:9200/tarena/nsd/1>

删除索引

curl –XDELETE <http://192.168.1.115:9200/tarena>

端口号后面/\* 代表删根，所以要放在内网中使用

curl –XDELETE [http://192.168.1.115:9200/\*](http://192.168.1.115:9200/*)

危险操作

ELK之二：kibana 安装使用

1. kibana的介绍：

数据可视化平台工具，可以让开发人员轻松查看日志，建议封装在docker中

2. kibana的安装：

1.安装java-1.8.0-openjdk 和 kibana

yum –y install java-1.8.0-openjdk kibana

2.修改kibana配置文件

vim /opt/kibana/config/kibana.yml

server.port: 5601

server.host: 0.0.0.0

elasticsearch.url: <http://es01:9200>

kibana.index: “.kibana”

elasticsearch.pingTimeout: 1500

elasticsearch.requestTimeout: 30000

elasticsearch.startupTimeout: 5000

kibana.defaultAppId: “discover”

其中index表示在elasticsearch中显示的索引名称

3.启动kibana

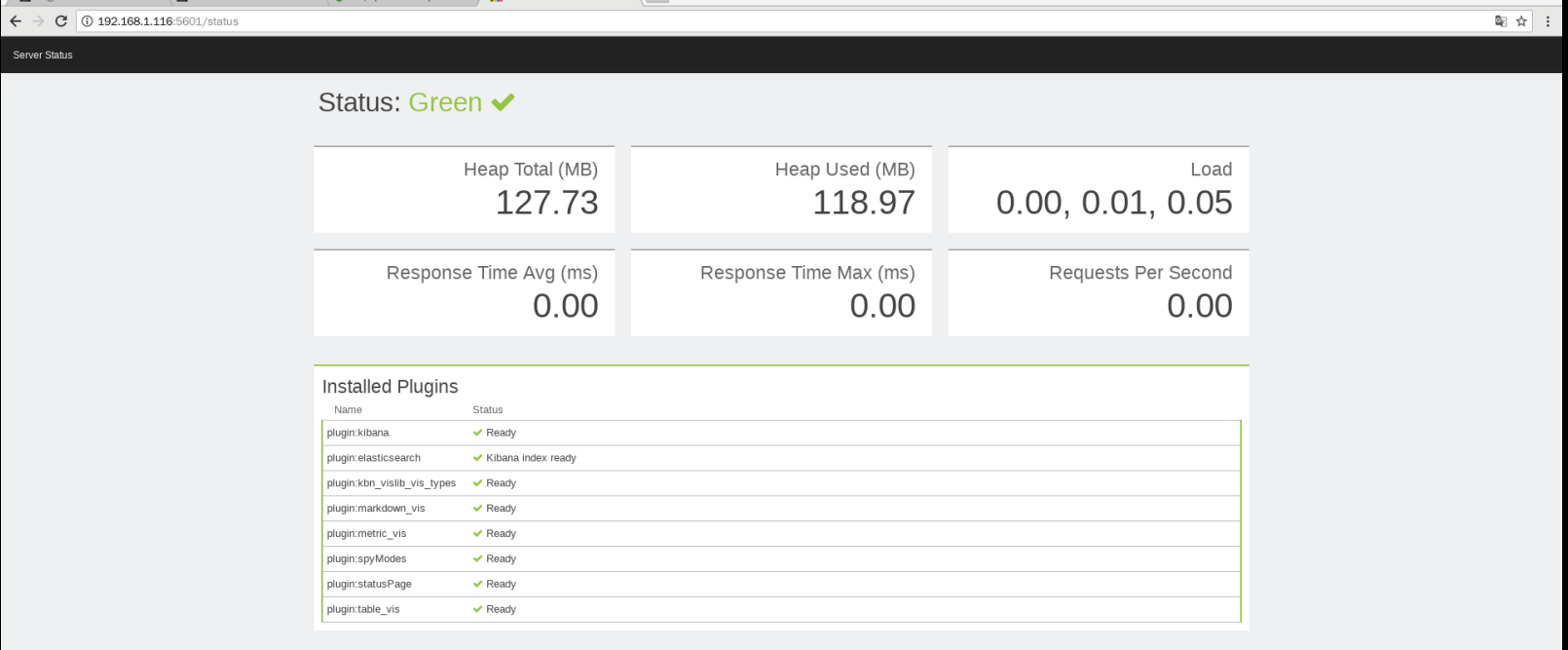
systemctl start kibana

此时5601端口被监听，在head页面中有.kibana索引，表示启动成功



<http://192.168.1.116:5601>

点击status标签页，状态为ok，下面都是√，则表示kibana搭建成功



1. kibana的使用
2. 数据批量导入
3. 使用\_bulk导入，调用POST方式
4. 数据格式：
   1. json 几行数据
   2. data-binary 大文件
5. 下载模板数据

lftp 192.168.1.254

cd elk

get shakespeare.json.gz

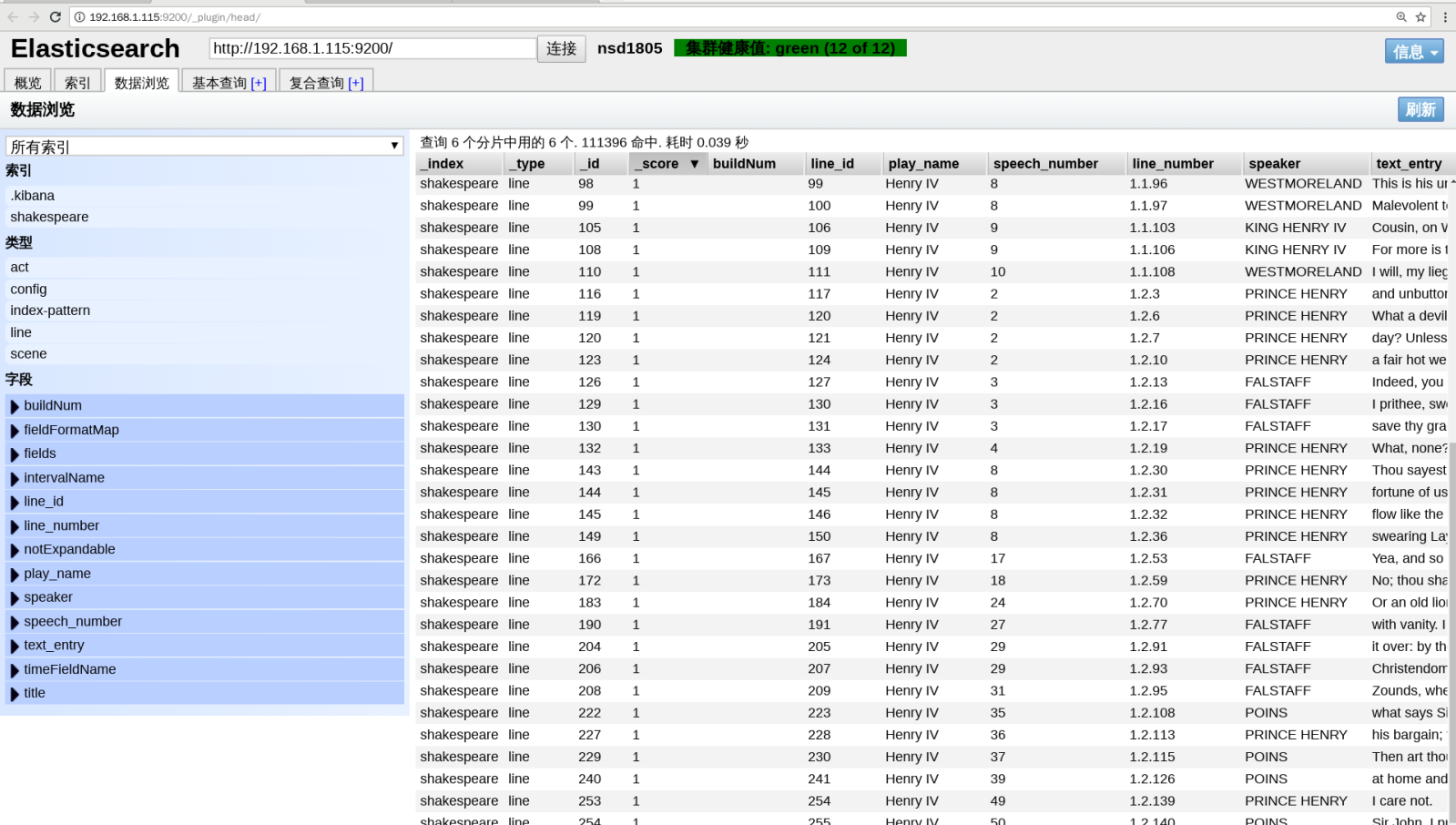
get log.jsonl.gz

get accounts.json.gz

1. 导入数据

curl -XPOST http://192.168.1.115:9200/\_bulk --data-binary @shakespeare.json

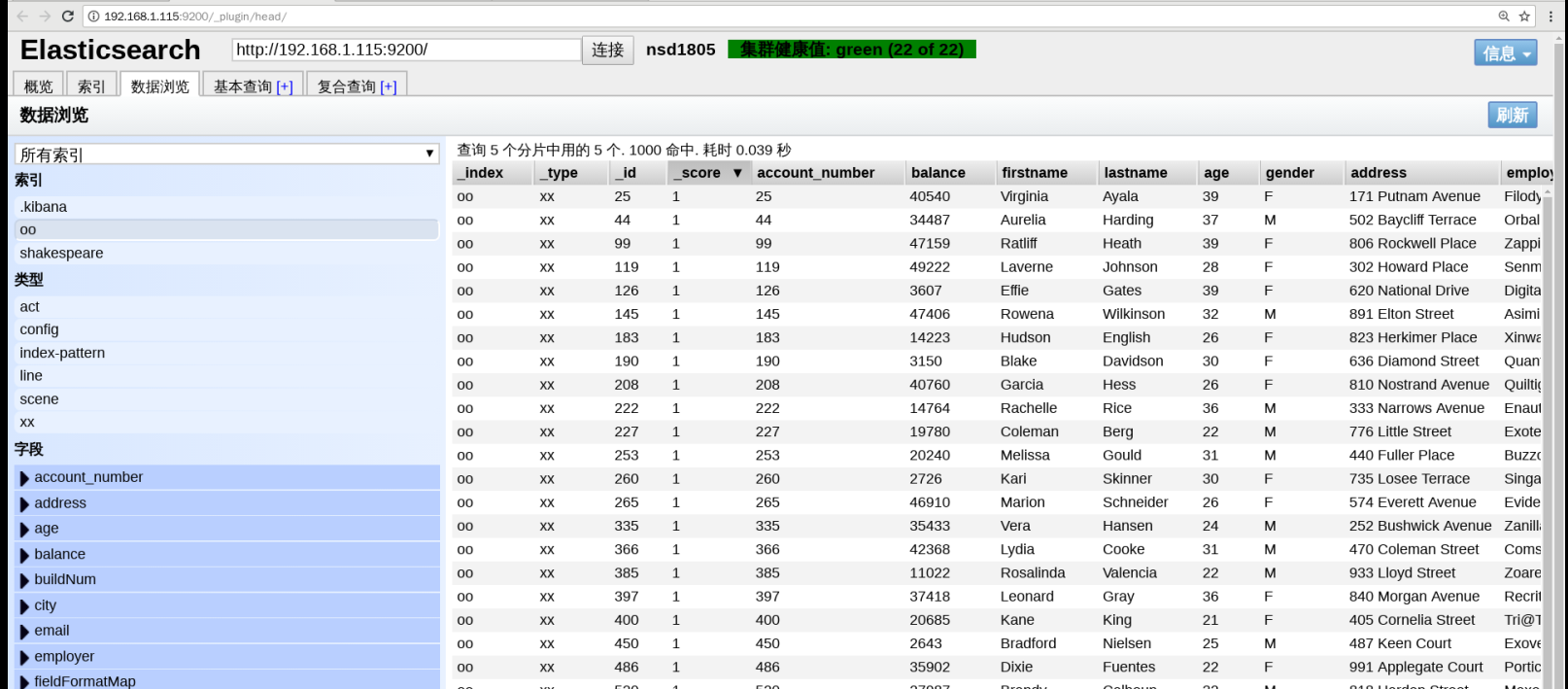
1. 查看数据是否导入成功



1. 继续导入和查看

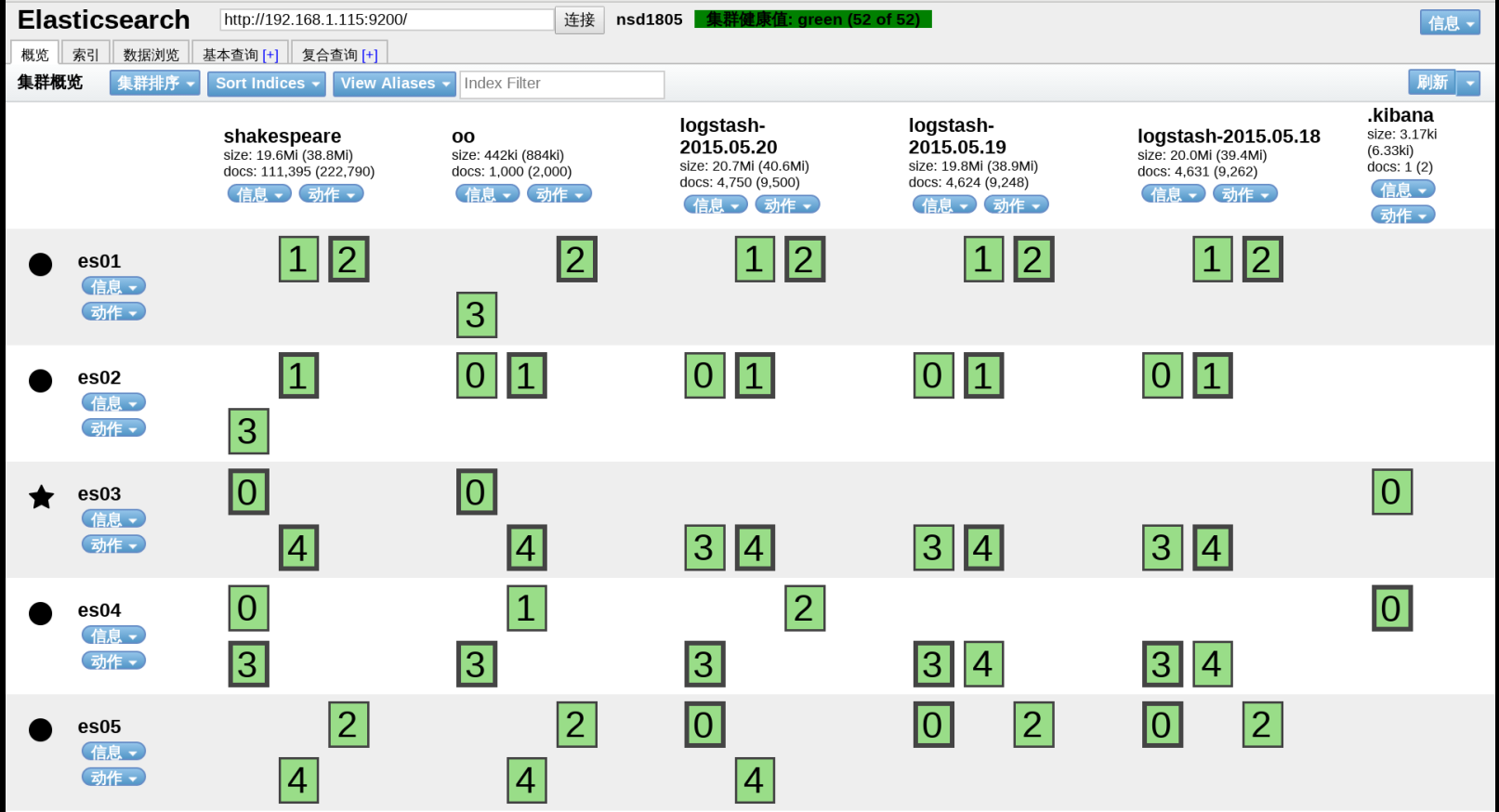
curl -XPOST http://192.168.1.115:9200/oo/xx/\_bulk --data-binary @accounts.json

两个数据表头不一样，第二个没有索引和类型，需要我们自己在http后加索引和类型



1. 导入第三个数据文件

curl -XPOST http://192.168.1.115:9200/\_bulk --data-binary @logs.json

2. 数据批量查询

数据批量查询需要加上\_mget选项

curl -XGET 'http://192.168.1.113:9200/\_mget?pretty' -d '{

"docs":[

{

"\_index":"accounts",

"\_type":"act",

"\_id":1

},

{

"\_index":"shakespeare",

"\_type":"line",

"\_id":14

}

]

}'

3. 安装web服务器，并启动

yum -y install httpd

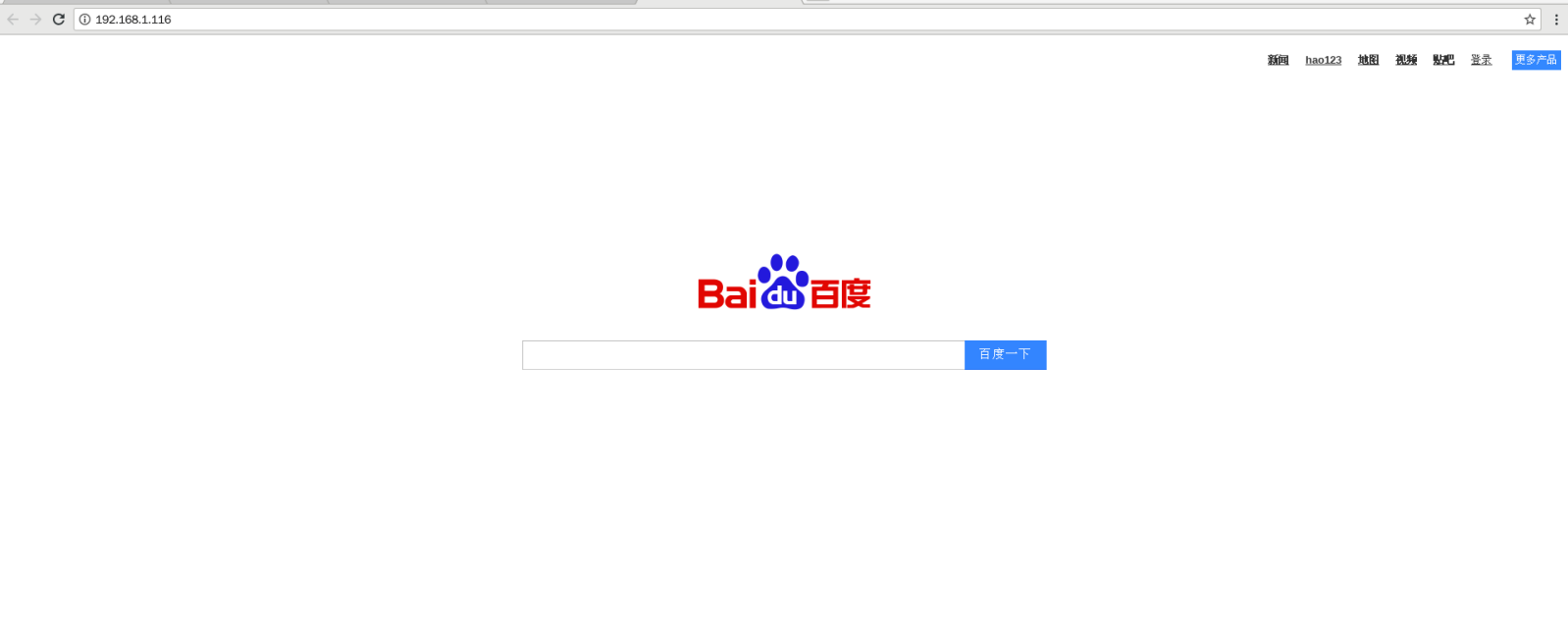
cd /var/www/html/

curl http://www.baidu.com -o index.html

vim /etc/httpd/conf/httpd.conf

systemctl restart httpd

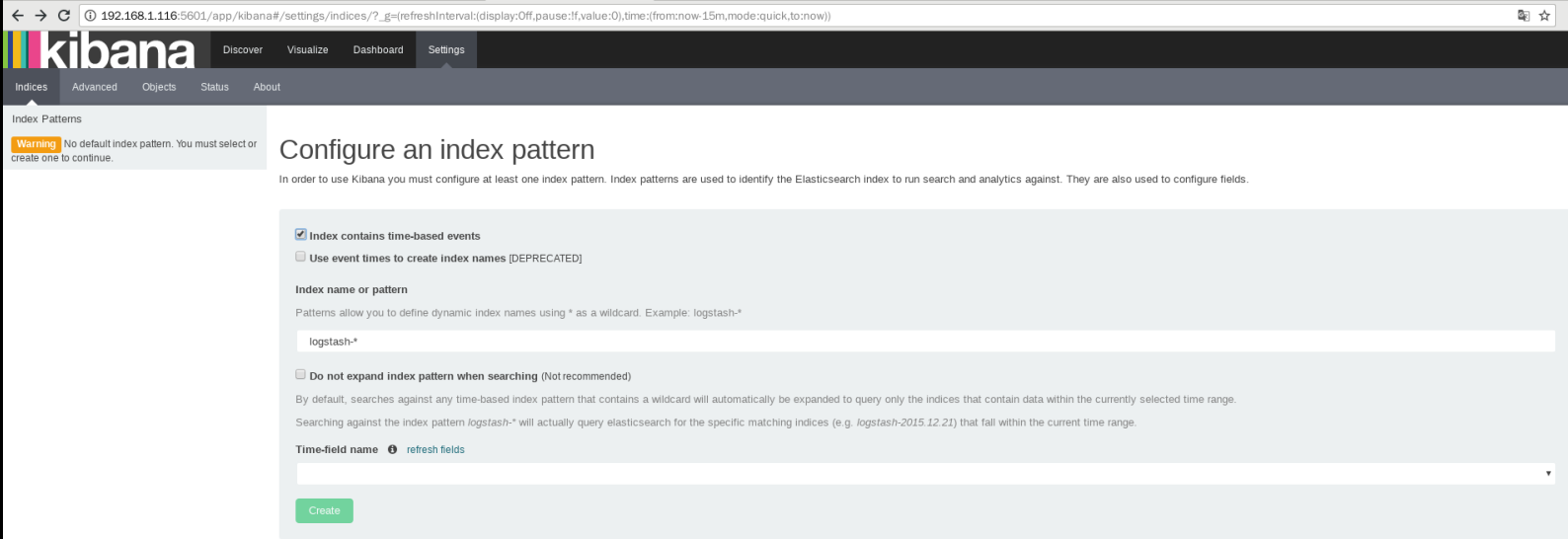
<http://192.168.1.116>



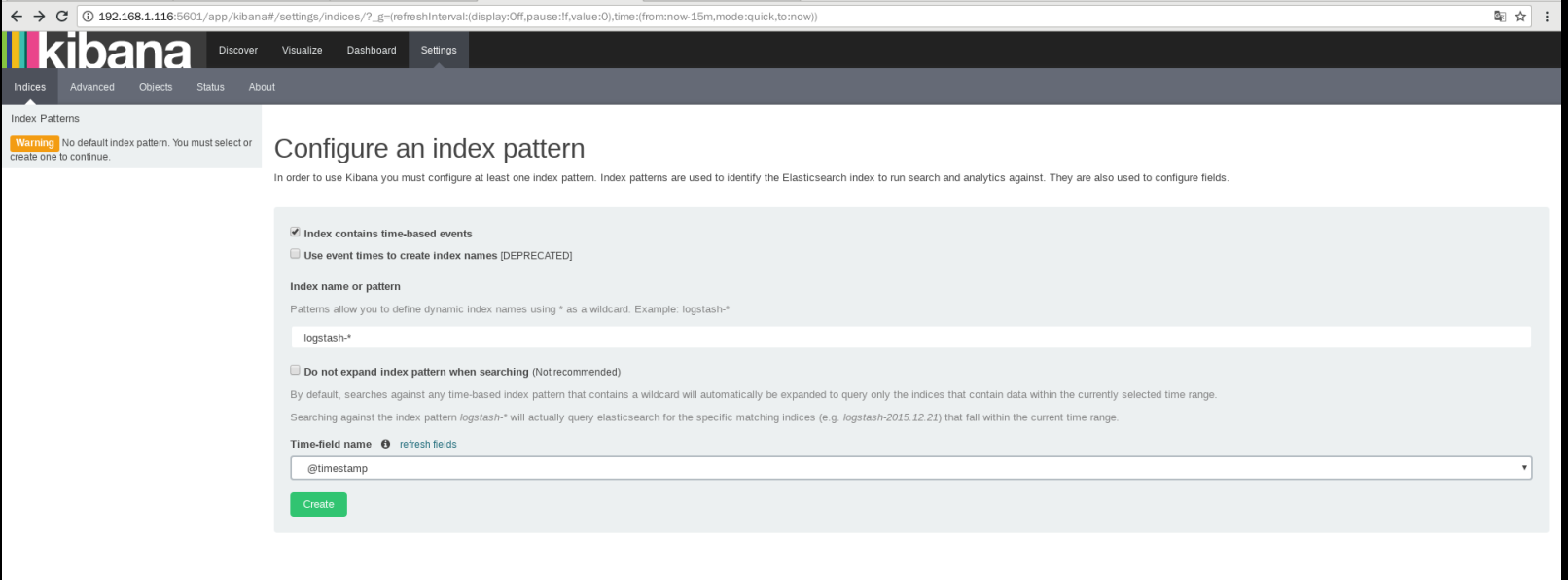
cat /etc/httpd/logs/access\_log

192.168.1.254 - - [22/Sep/2018:16:27:13 +0800] "GET / HTTP/1.1" 200 2381 "-" "Mozilla/5.0 (X11; Linux x86\_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/60.0.3112.113 Safari/537.36"

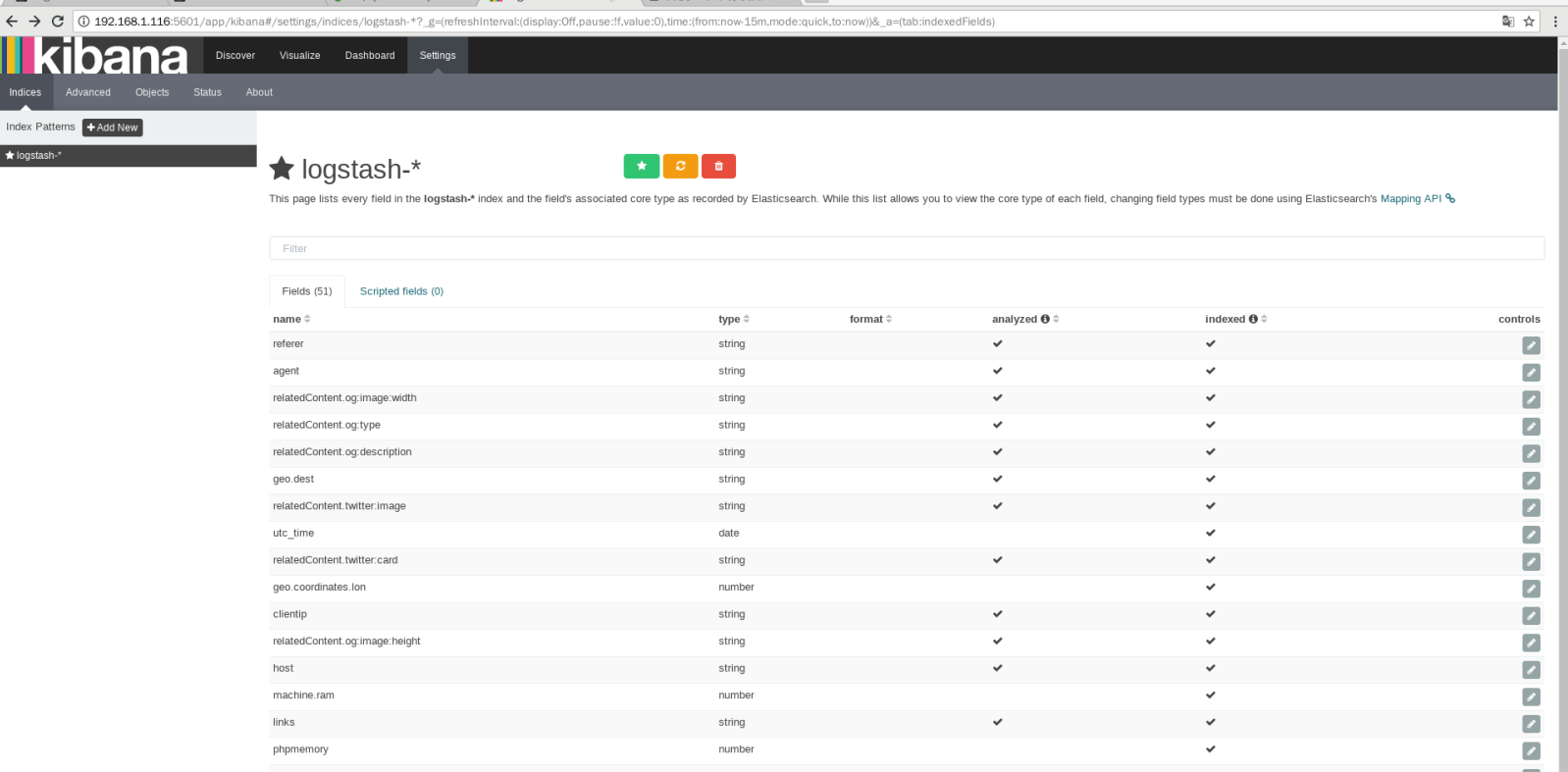
4. kibana页面配置



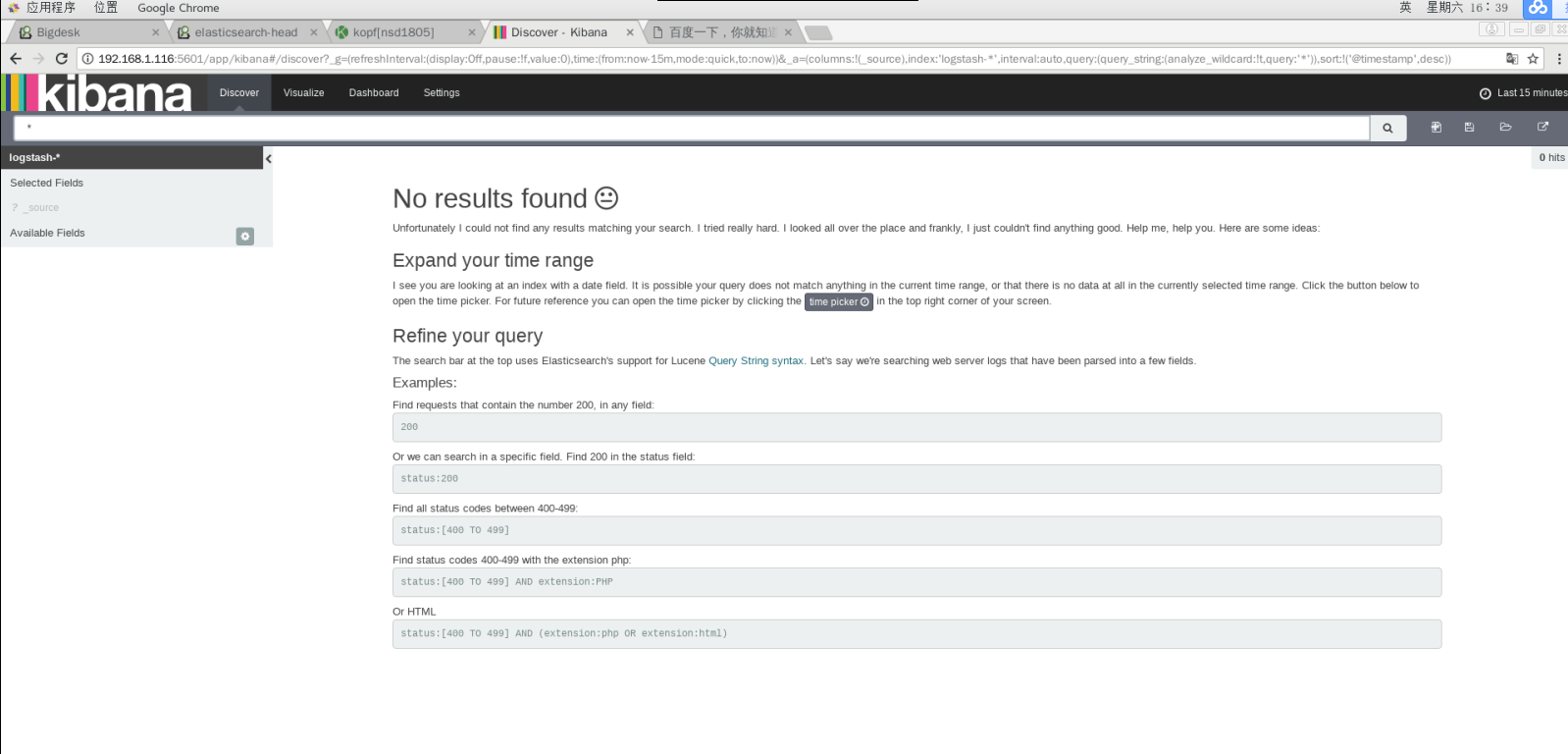
默认会有warning小警告，需要进行配置



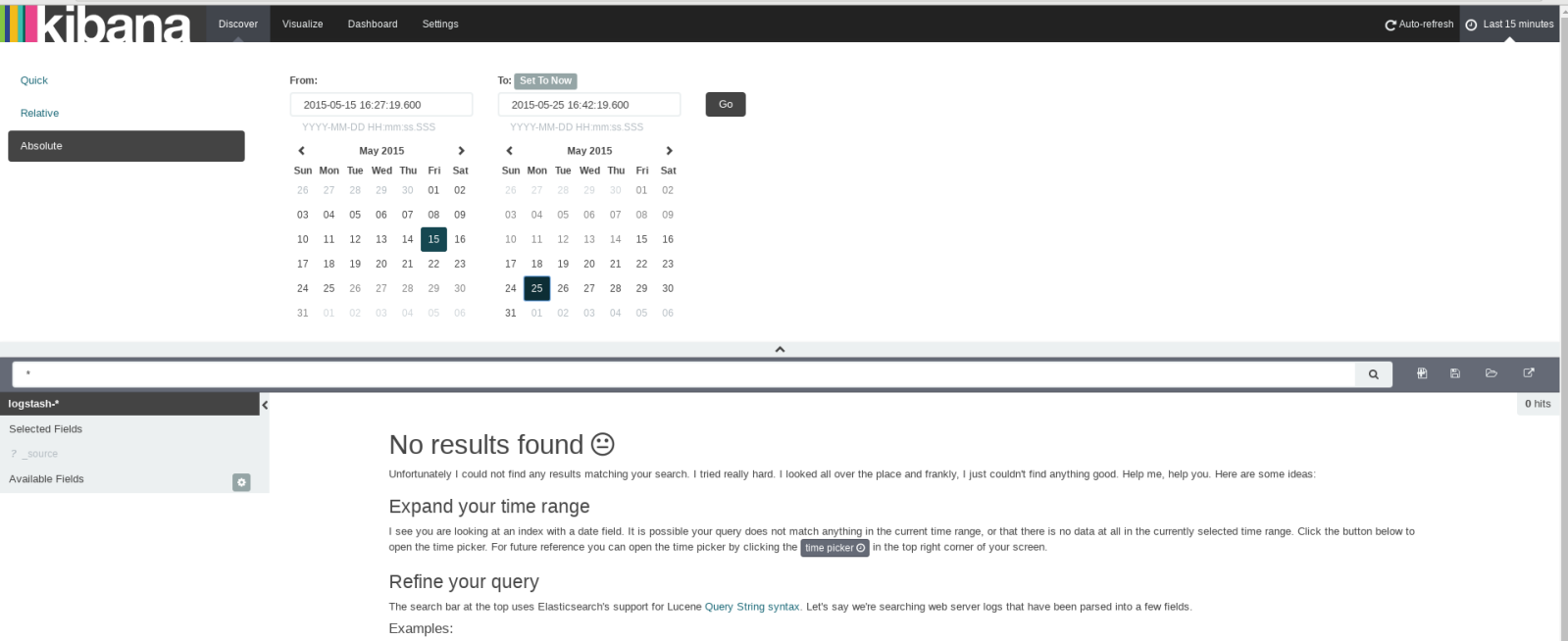
选择@timestamp，create，就没有warning了



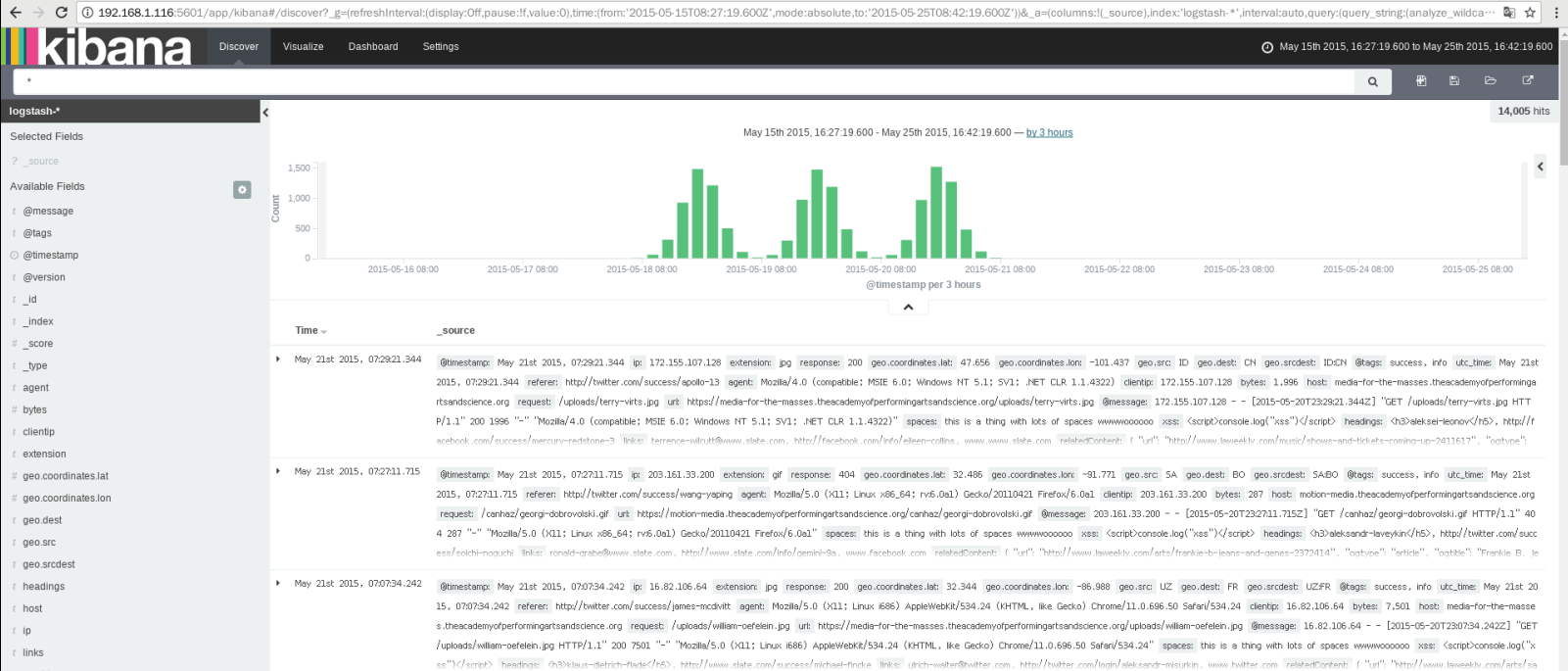
选择上面discover按钮



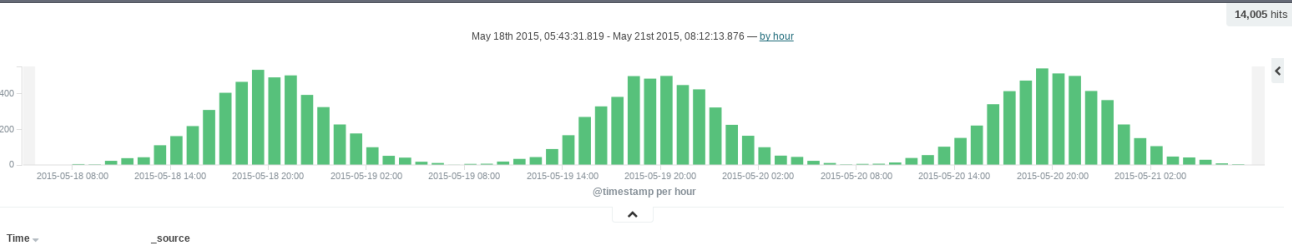
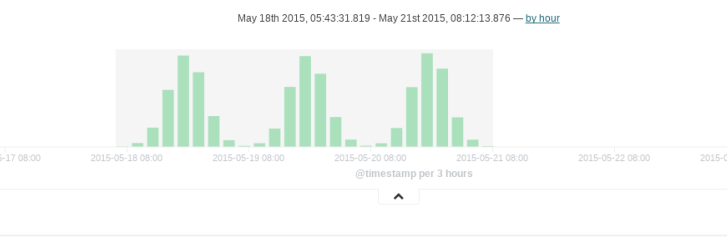
要选择时间段，才能显示想要的数据【kibana默认只识别最近15分钟内容】



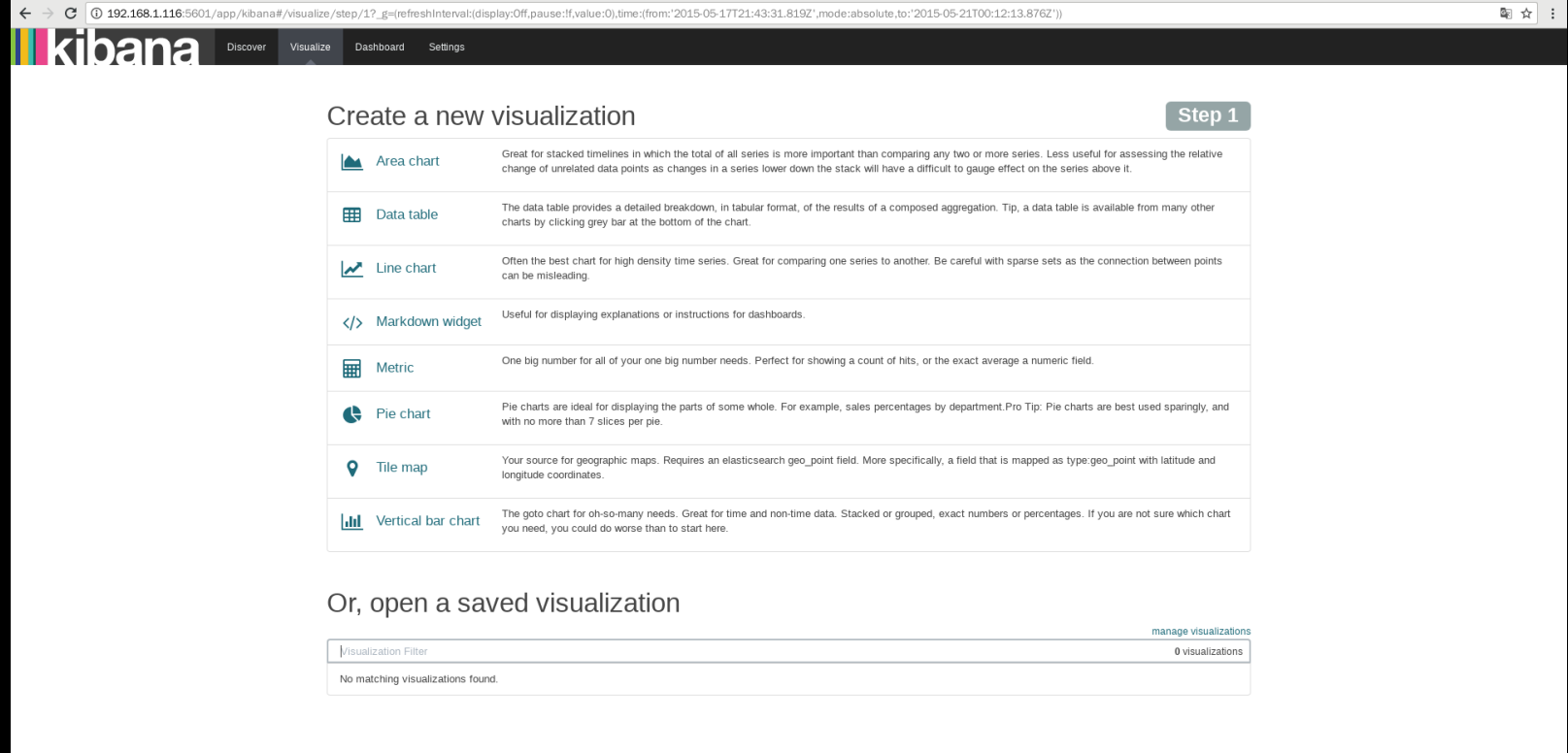
时间选对了，数据就出来了



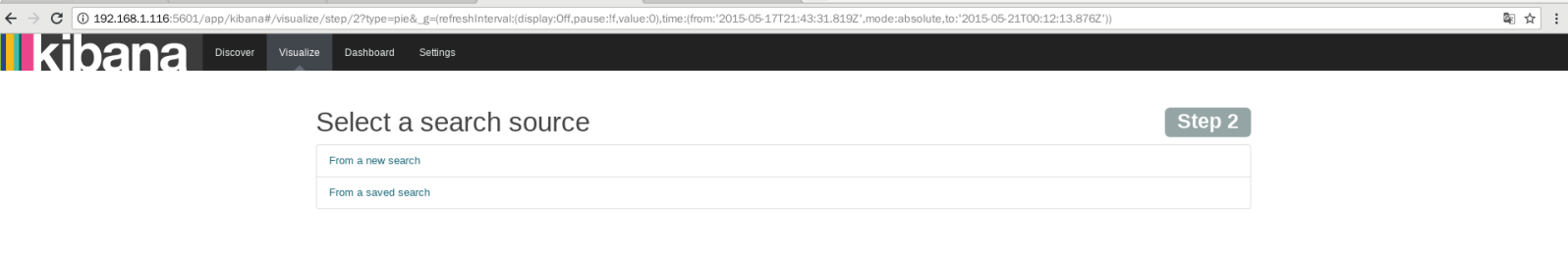
画框选中则更精确



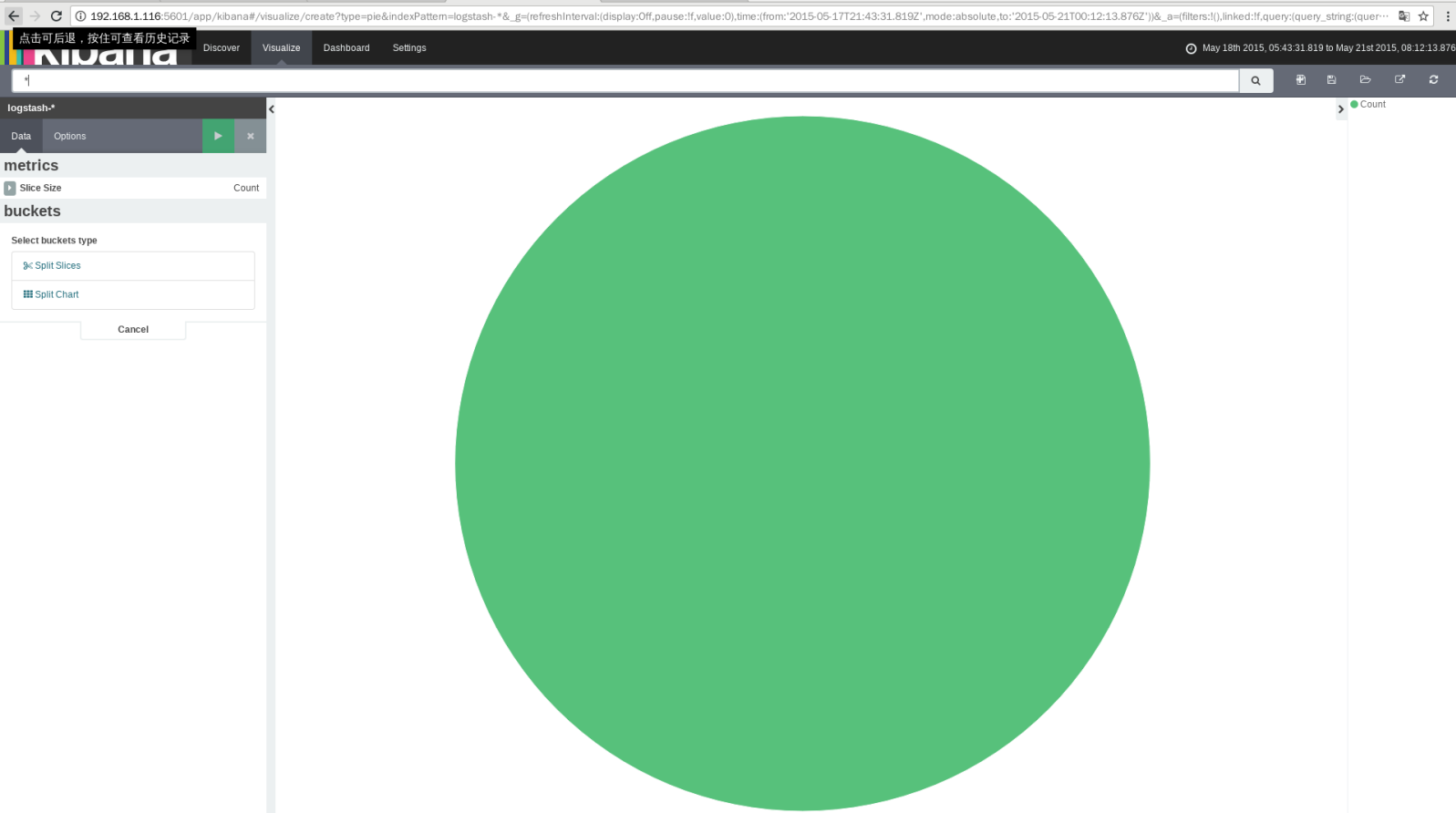
选择visualize标签页，进行绘制其它图形



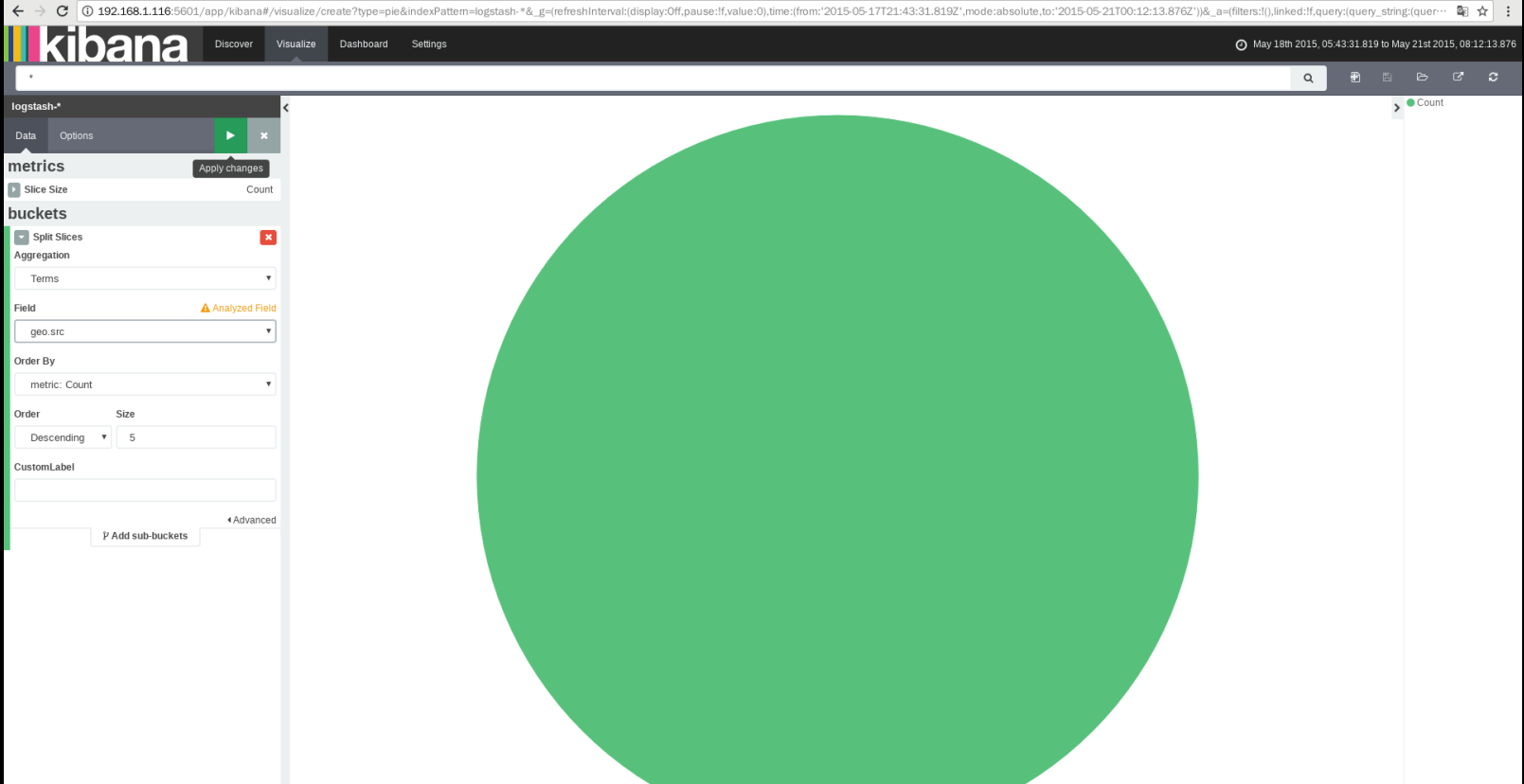
选择饼状图，创建并状图



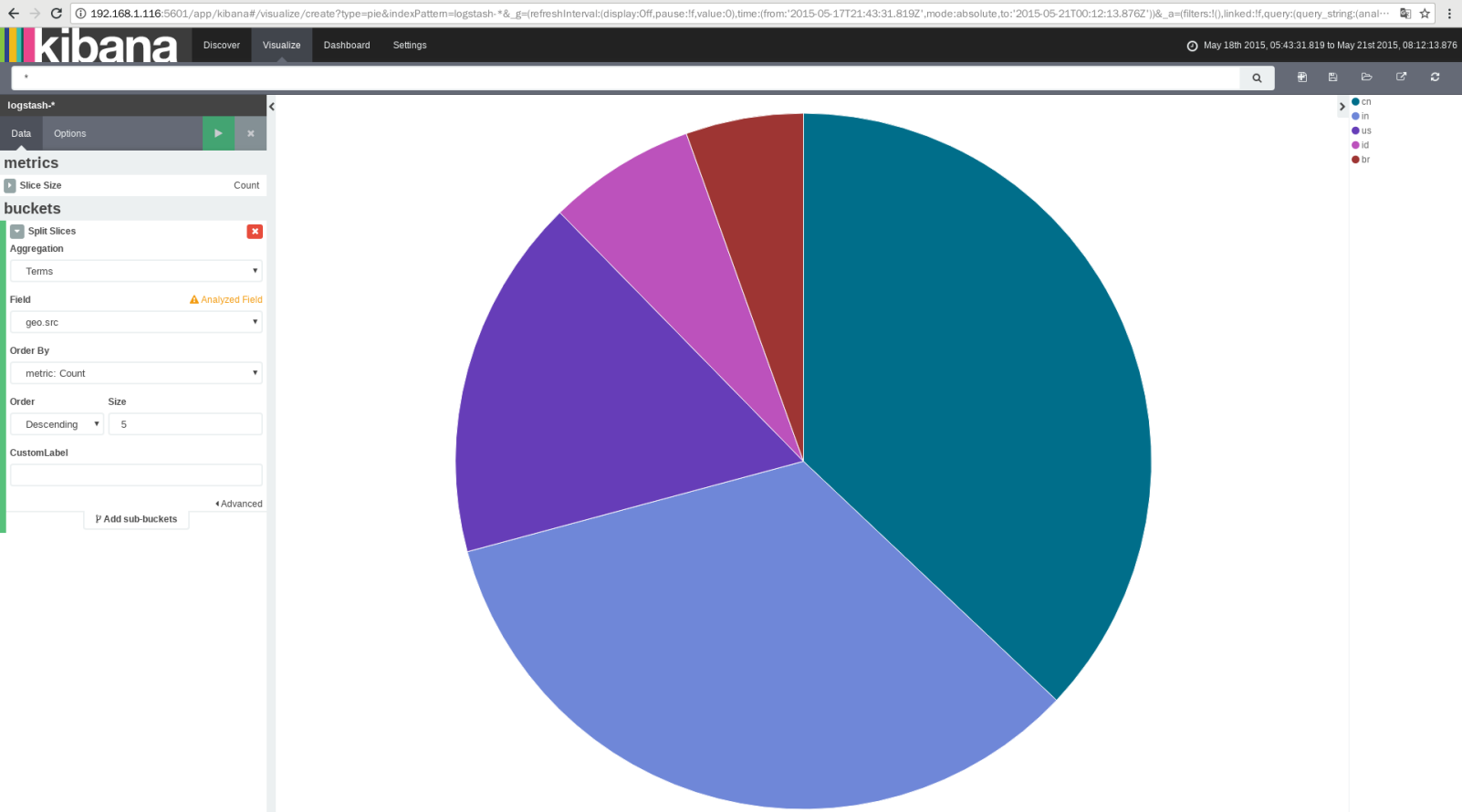
选择from a new search



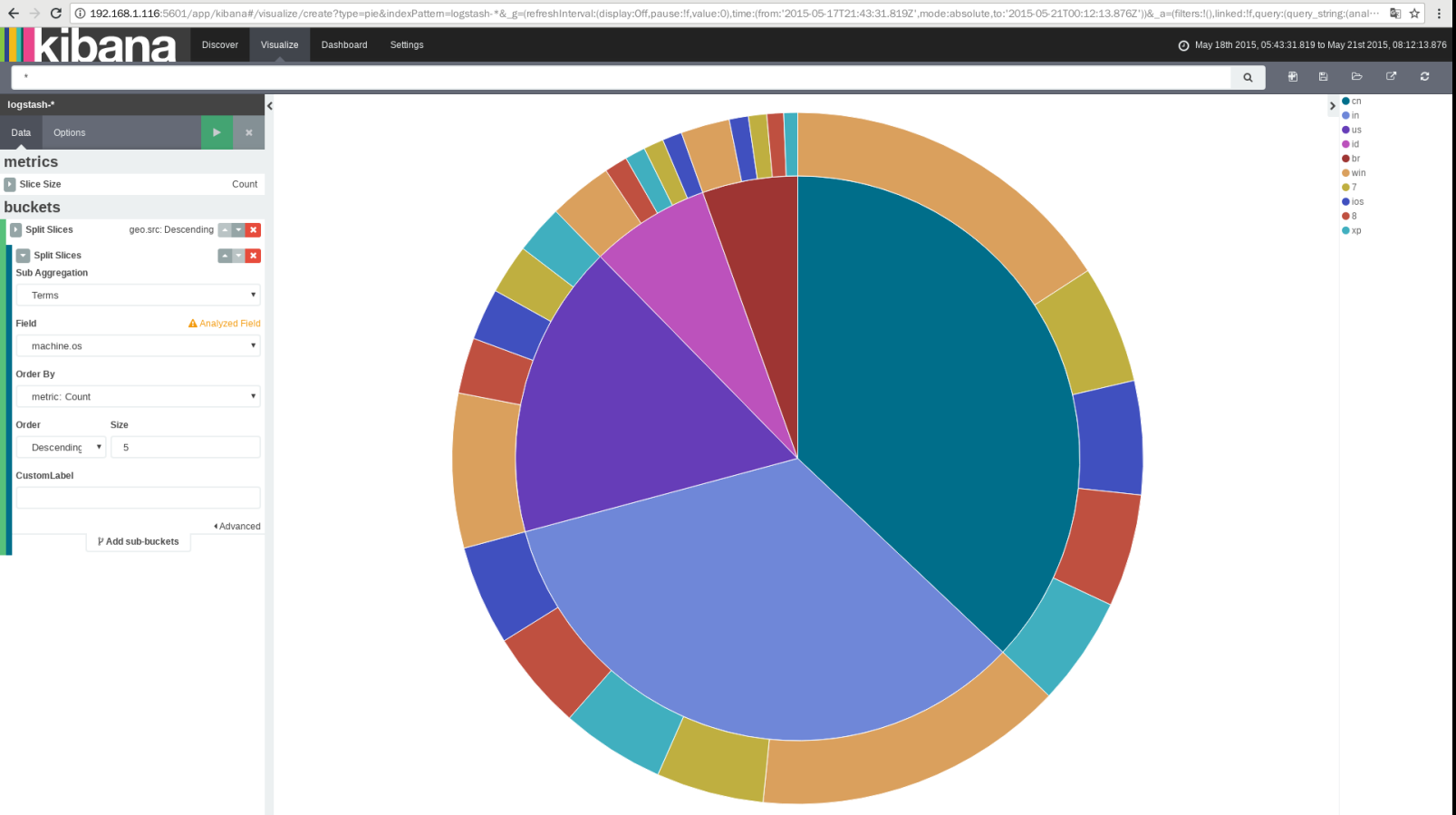
选择 split slices 第一个



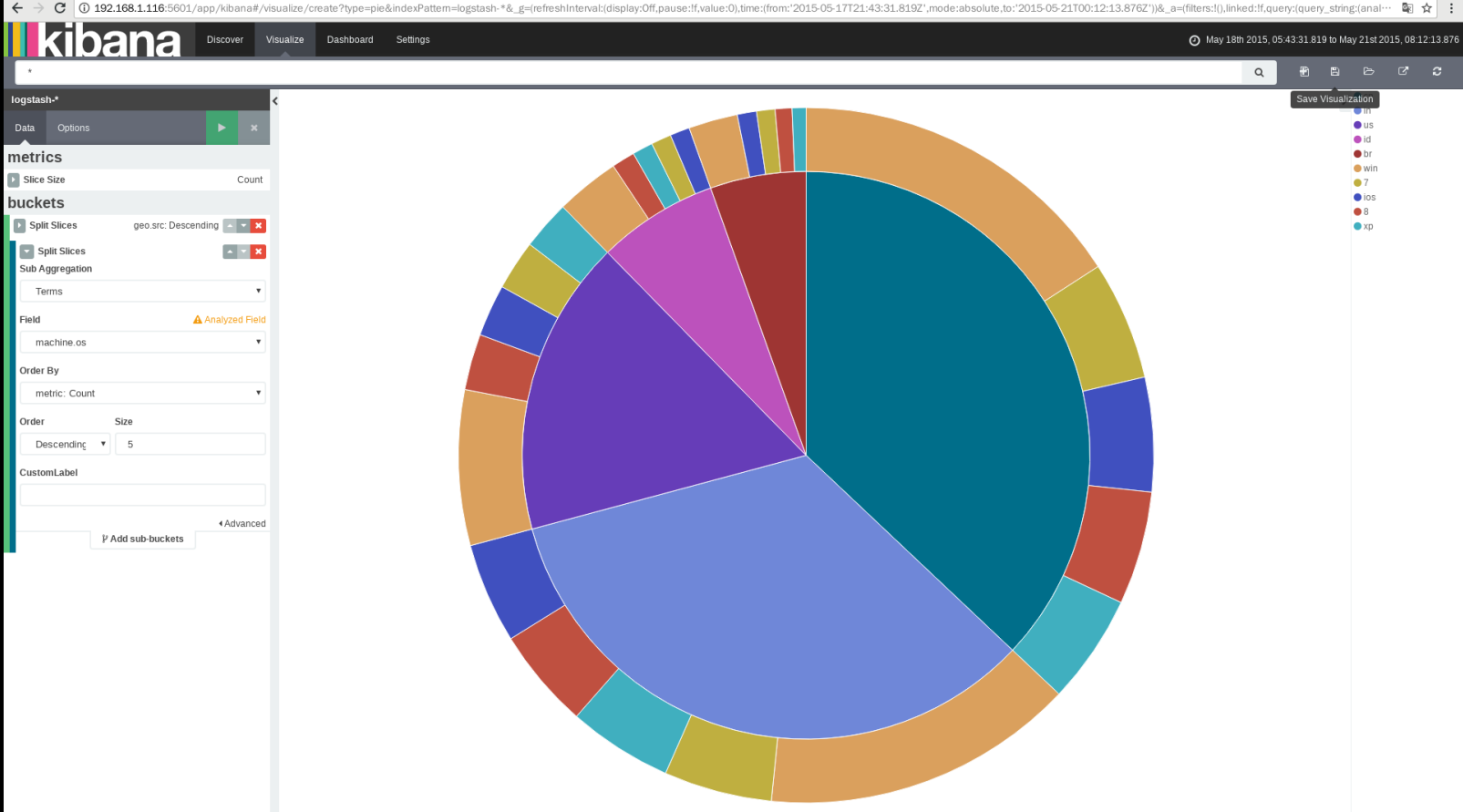
按播放按钮

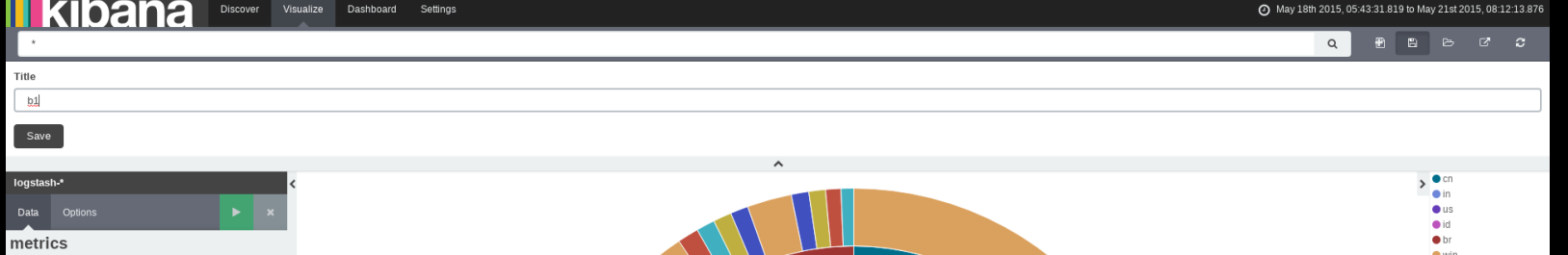


还可以添加子选项

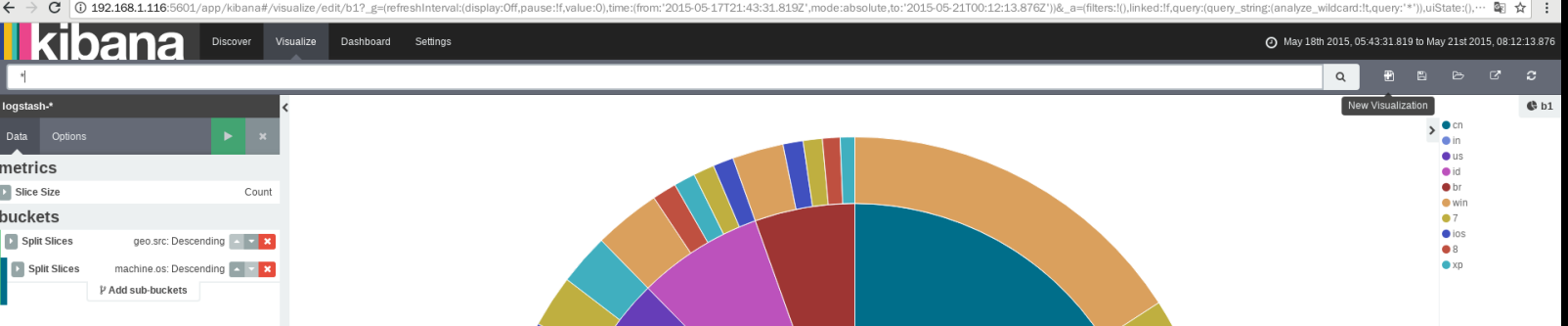


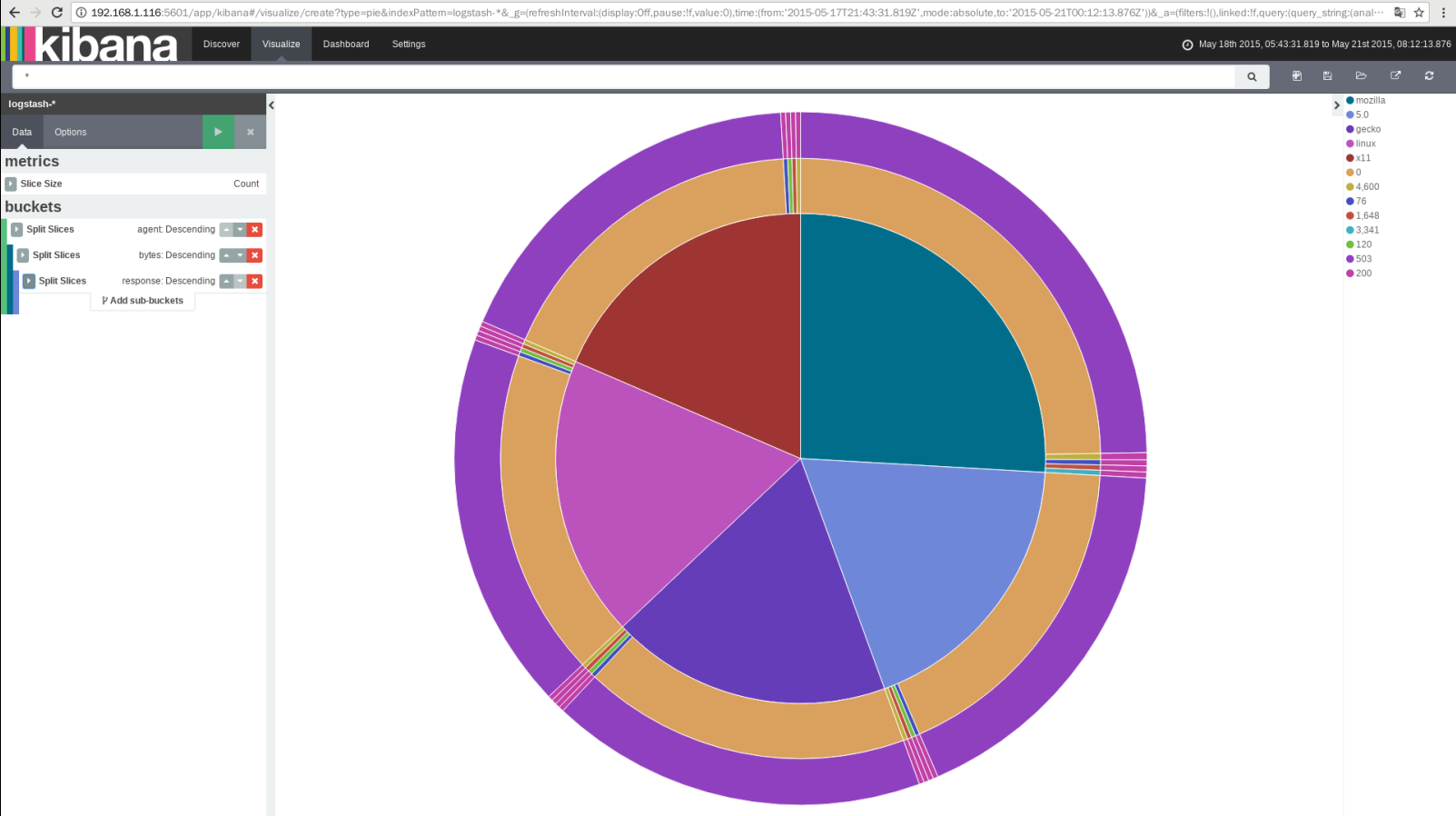
做完后保存，网页右上角有保存按钮



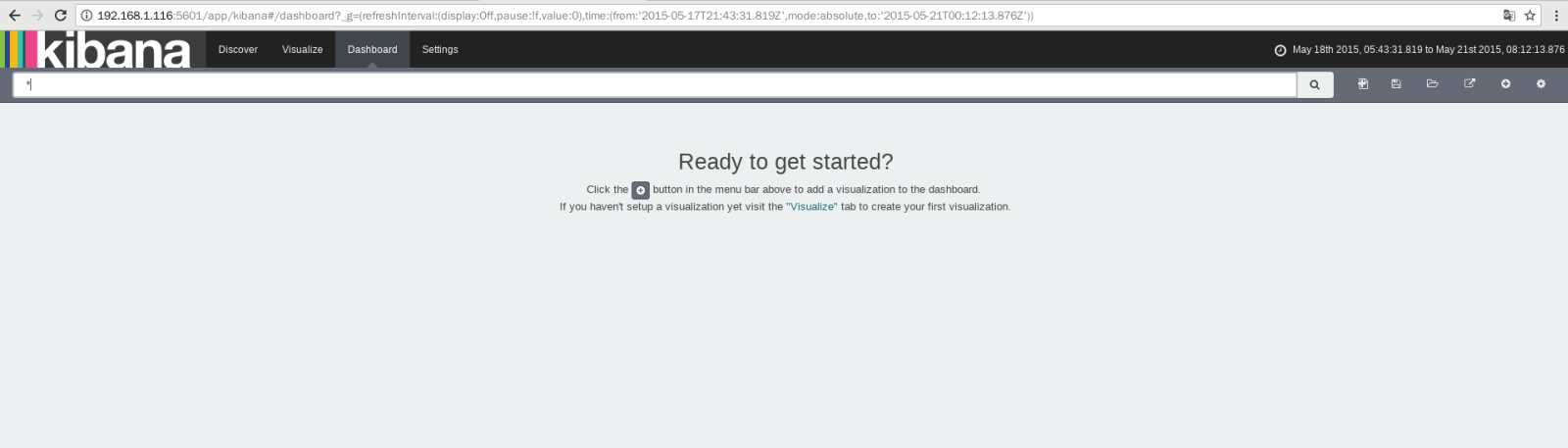


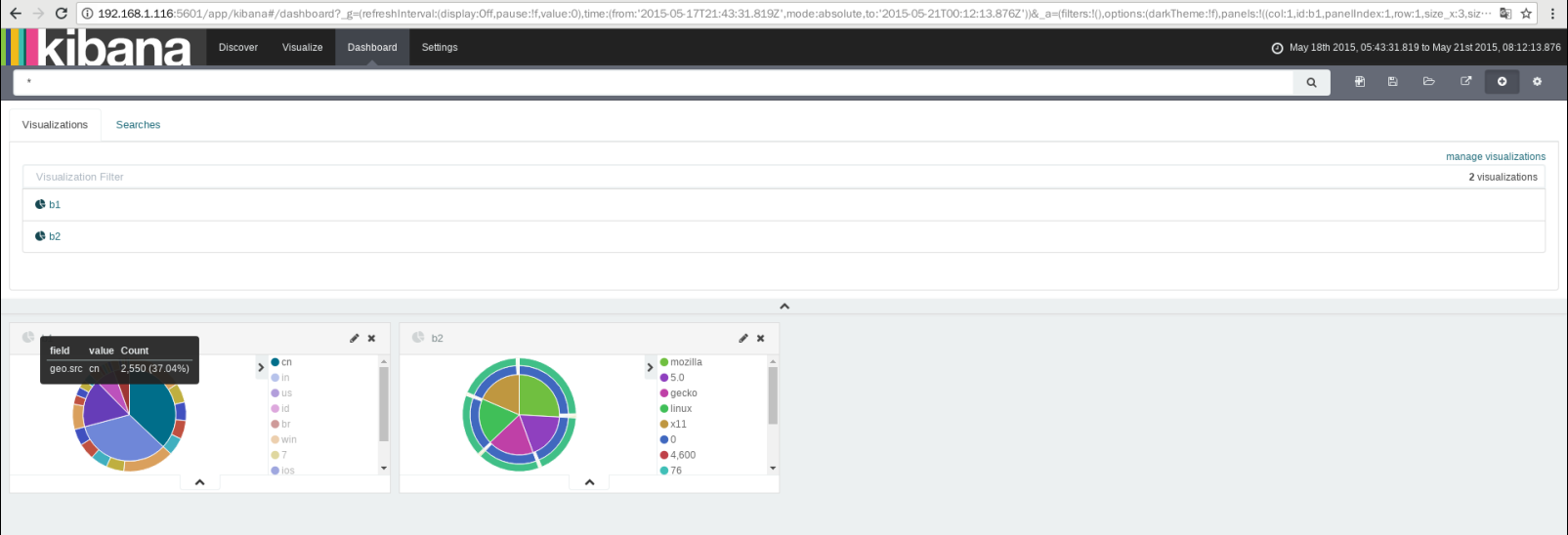
右上角新建，继续创建



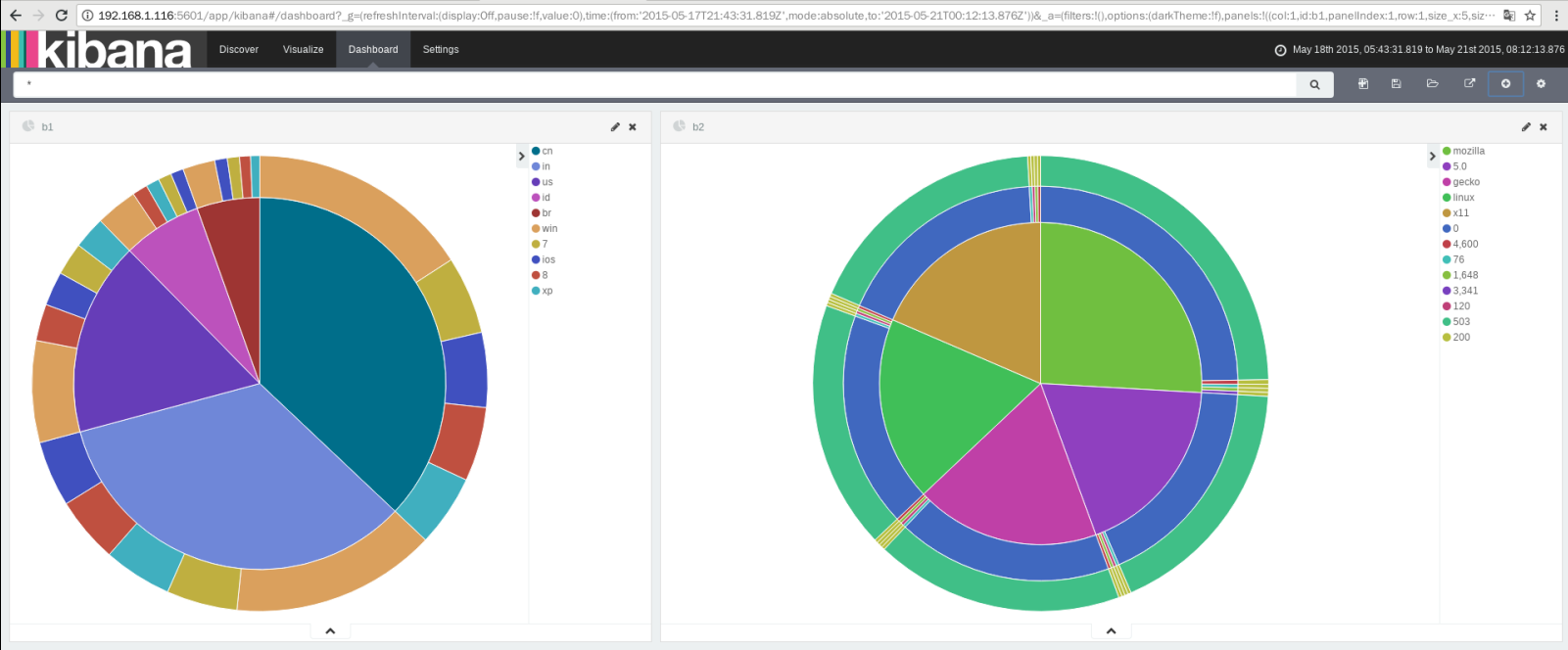


选择dashboard，右侧有+号，取出两张图





可以放大缩小



ELK之三：logstash 安装使用

1.配置主机名，ip和yum源，配置/etc/hosts

Vim /etc/hosts

192.168.1.110 es01

192.168.1.112 es02

192.168.1.113 es03

192.168.1.114 es04

192.168.1.115 es05

192.168.1.116 kibana

192.168.1.120 logstash

2.安装openjdk和logstash

Yum –y install java-1.8.0-openjdk logstash

Java –version

Touch /etc/logstash/logstash.conf

/opt/logstash/bin/logstash-plugin list

Vim /etc/logstash/logstash.conf

Input{

Stdin {codec => “json”}

}

Filter{}

Output{

Stdout {codec=>”rubydebug”}

}

/opt/logstash/bin/logstash –f /etc/logstash/logstash.conf

3.logstash-input各个模块讲解

1. file模块插件

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

}

}

2.tcp、udp模块插件

Input{

tcp {

host => "0.0.0.0"

port => "8888"

type => "tcplog"

}

udp {

port => "9999"

type => "udplog"

}

}

测试脚本：vim sendmsg.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<&-

}

Chmod +x sendmsg.sh

Sendmsg udp “xx”

Sendmsg tcp “yy”

3.syslog模块插件

Input{

syslog {

port => "514"

type => "syslog"

}

}

4.把登录日志的信息发送给logstash

Vim /etc/rsyslog.conf

Authpriv.\* @@192.168.1.120:514

Systemctl restart rsyslog

5.收集自定义日志发送给logstash

Vim /etc/rsyslog.conf

Local0.\* @192.168.1.120:514

Systemctl restart rsyslog

Logger –p local0.info –t nsd “hello world”

6.收集本机自定义日志

Vim /etc/rsyslog.conf

Local0.\* /var/log/mylog

Systemctl restart rsyslog

Logger –p local0.info –t nsd “hello world”

4.logstash-filter模块讲解

1.grok插件

filter{

grok{

match => ["message", "(?<key>reg)"]

}

}

查找正则宏路径

ls /opt/logstash/vendor/bundle/jruby/1.9/gems/logstash-patterns-core-2.0.5/patterns/grok**-**patterns //查找COMBINEDAPACHELOG

filter{

grok{

match => ["message", "%{COMBINEDAPACHELOG}"]

}

}

5. 用filebeat收集Apache服务器的日志，存入elasticsearch

yum -y install filebeat

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: ["192.168.1.120:5044"]     //去掉注释,logstash那台主机的ip

[root@logstash ~]# vim /etc/logstash/logstash.conf

input{

stdin{ codec => "json" }

beats{

port => 5044

}

}

filter{

if [type] == "apachelog"{

grok{

match => ["message", "%{COMBINEDAPACHELOG}"]

}}

}

output{

stdout{ codec => "rubydebug" }

if [type] == "apachelog"{

elasticsearch {

hosts => ["192.168.1.110:9200", "192.168.1.112:9200"]

index => "apachelog"

flush\_size => 2000

idle\_flush\_time => 10

}}

}

[root@logstash ~]# netstat -antup | grep 5044

浏览器访问Elasticsearch，有apachelog