

# Logash+Elasticsearch+Kibana 日志系统安装部署

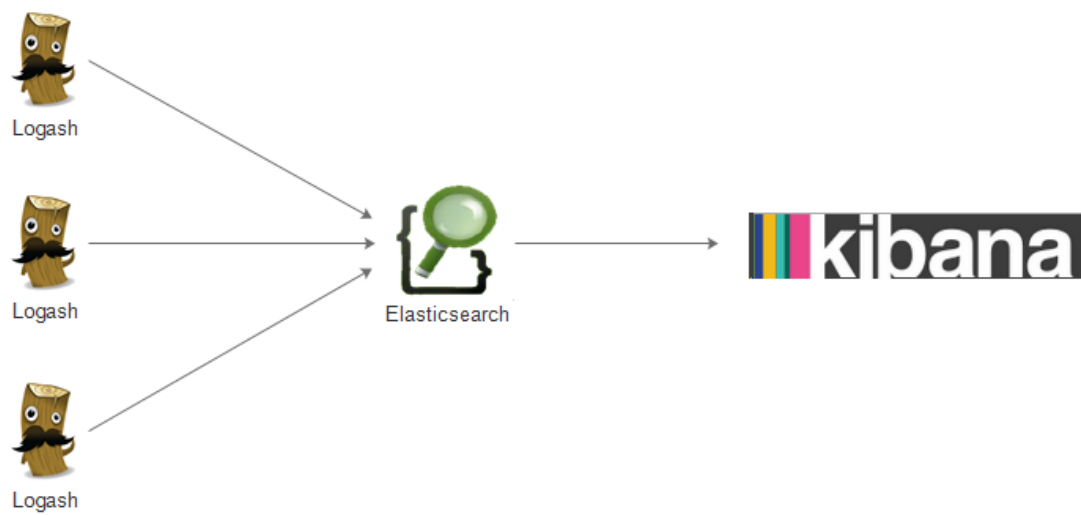
网海过客 [www.chinasa.net](http://www.chinasa.net)

## 环境

类型	操作系统	IP	软件包		
服务端	Centos 6.5 X64	192.168.0.100	JDK1.8	Elasticsearch-1.4.4	Kibana-4.0.2
客户端	Centos 6.5 X64	192.168.0.101	JDK1.8	Logash-1.4.2	

**注：** Logash、Elasticsearch、Kibana 运行需要 JDK。

Logash Elasticsearch Kibana 流程图



## 服务端安装-192.168.0.100

### JDK 安装

```
#yum install java-1.8.0-openjdk
```

### Elasticsearch 安装

```
#tar zxvf elasticsearch-1.4.4.tar.gz -C /opt
```

配置

```
#vi /opt/elasticsearch-1.4.4/conf/elasticsearch.yml
```

加入下面内容:

```
http.cors.enabled: true
```

```
http.cors.allow-origin: "*"
```

注: 默认端口:9200

### Elasticsearch 启动

```
#!/opt/elasticsearch-1.4.4/bin/elasticsearch -Xmx2g -Xms2g -Des.index.storage.type=memory -d
```

注: -Xmx2g 为最小内存和最大内存

-d 后台运行

## Kibana 安装

```
#tar zxvf kibana-4.0.2-linux-x64.tar.gz -C /opt
```

配置

```
#vi /opt/logash/kibana-4.0.2-linux-x64/config/kibana.yml
```

添加 elasticsearch 地址，添加即可。

elasticsearch\_url: <http://localhost:19200>

## Kibana 启动

```
#!/opt/logash/kibana-4.0.2-linux-x64/bin/kibana -p 25601 > /dev/null 2>&1 &
```

**注:** -p 指定端口，默认端口 5601

## 客户端安装-192.168.0.101

### JDK 安装

```
#yum install java-1.8.0-openjdk
```

### Logash 安装

```
#tar zxvf logstash-1.4.2.tar.gz -C /opt
```

### Logash 配置

创建 nginx\_log.conf 配置文件

```
#vi /opt/logstash-1.4.2/conf/nginx_log.conf
```

内容如下:

```
input {  
    file {  
        type => "nginx_log"  
        path => "/opt/nginx/logs/access.log"  
    }  
}  
output {  
    stdout { codec => rubydebug }  
    elasticsearch_http {  
        host => "192.168.0.100"  
        port => "9200"  
    }  
}
```

创建 nginx 日志格式配置文件

**注:** path => "/opt/nginx/logs/access.log" #Nginx 日志文件  
match => { "message" => "%{NGINXACCESS}" } #Nginx 日志格式, 变量%{NGINXACCESS}会自动在/opt/logstash-1.4.2/patterns 目录下查找。

host => "192.168.0.100" # elasticsearch 服务端 IP  
port => "9200" # elasticsearch 服务端端口

```
#vi /opt/logstash-1.4.2/patterns/nginx
NGUSERNAME [a-zA-Z\.\@\-\+\_%]+
NGUSER %{NGUSERNAME}
NGINXACCESS %{IPORHOST:remote_addr} - - \[%{HTTPDATE:time_local}\]
"%{WORD:method} %{URIPATHPARAM:request}
HTTP/%{NUMBER:httpversion}" %{INT:status} %{INT:body_bytes_sent} %{QS:http_referer} %{QS:
http_user_agent}
```

## Logash 启动

```
#/opt/logstash-1.4.2/bin/logstash -f /opt/logstash-1.4.2/conf/nginx_log.conf > /dev/null 2>&1 &
```

## Kibana 添加地图功能

```
#wget http://geolite.maxmind.com/download/geoip/database/GeoLiteCity.dat.gz
#gunzip GeoLiteCity.dat.gz
#mv GeoLiteCity.dat /opt/logstash-1.4.2/conf
```

在/opt/logstash-1.4.2/conf/nginx\_log.conf 配置文件 **filter{}** 加入以下内容:

```
geoip {
  source => "remote_addr"
  target => "geoip"
  database => "/opt/logstash-1.4.2/conf/GeoLiteCity.dat"
  add_field => [ "[geoip][coordinates]", "%{[geoip][longitude]}" ]
  add_field => [ "[geoip][coordinates]", "%{[geoip][latitude]}" ]
}
mutate {
  convert => [ "[geoip][coordinates]", "float" ]
}
}
```

**注:** source => "remote\_addr" #对应 nginx 日志客户端 IP

## Kibana 添加高德地图

修改 index.js 文件

```
#vi kibana-4.0.2-linux-x64/src/public/index.js
```

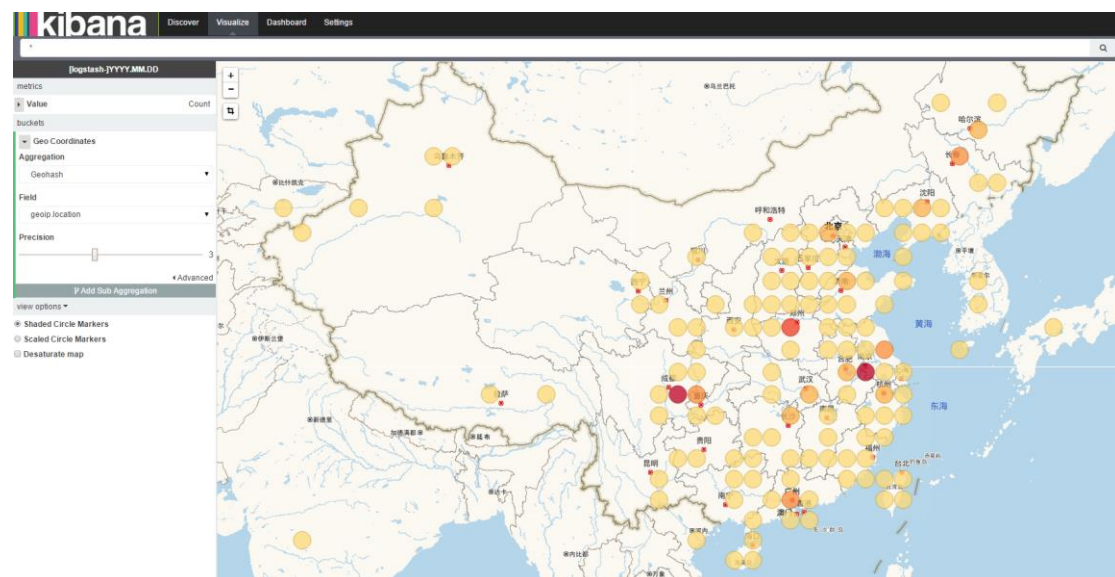
```
/*      var tileLayer = L.tileLayer('https://otile{s}-s.mqcdn.com/tiles/1.0.0/map/{z}/{x}/{y}.jpeg', {
      attribution: 'Tiles by <a href="http://www.mapquest.com/">MapQuest</a> &mdash; ' +
      'Map data &copy; <a href="http://openstreetmap.org">OpenStreetMap</a> contributors, ' +
      '<a href="http://creativecommons.org/licenses/by-sa/2.0/">CC-BY-SA</a>',
      subdomains: '1234'
    });
  */
```

```
*/
```

将以上内容注释，并添加以下内容：

```
/*
  gaode map
*/
var tileLayer =
L.tileLayer('http://webd0{s}.is.autonavi.com/appmaptile?lang=zh_cn&size=1&scale=1&style=8&x={x}&y={y}&z={
z}', {
  attribution: 'Tiles by <a href="http://www.mapquest.com/">MapQuest</a> — ' +
  'Map data ? <a href="http://openstreetmap.org">OpenStreetMap</a> contributors, ' +
  '<a href="http://creativecommons.org/licenses/by-sa/2.0/">CC-BY-SA</a>',
  subdomains:['1","2","3","4"],
  variants: {
    Satellite:{
      url: 'http://webst0{s}.is.autonavi.com/appmaptile?style=6&x={x}&y={y}&z={z}'
    }
  }
});
```

高德地图效果



## Elasticsearch 认证

使用 Shield 对 Elasticsearch 认证

安装

```
#/opt/logash/elasticsearch-1.4.4/bin/plugin -i elasticsearch/license/latest
```

```
#/opt/logash/elasticsearch-1.4.4/bin/plugin -i elasticsearch/shield/latest
```

添加用户，并设置密码

```
#/opt/logash/elasticsearch-1.4.4/bin/shield/esusers useradd es_admin -r admin
```

Enter new password:

Retype new password:

**注:** elasticsearch 设置了用户名密码，logash 客户端也需要添加用户名密码，才能将日志传入 elasticsearch 里。

Kibana 也需要设置用户名密码，才能正常访问

如需要取消认证，直接将 shield 删除

```
#/opt/logash/elasticsearch-1.4.4/bin/plugin -r elasticsearch/license/latest
```

```
#/opt/logash/elasticsearch-1.4.4/bin/plugin -r elasticsearch/shield/latest
```

**注:**取消了 elasticsearch 认证，需要将 kibana 和 logash 客户端配置文件取消用户名密码配置。

## Kibana 认证

修改配置文件

```
#vi /opt/logash/kibana-4.0.2-linux-x64/config/kibana.yml
```

添加用户名密码

```
kibana_elasticsearch_username: es_admin
```

```
kibana_elasticsearch_password: 123456
```

## Logstash 客户端添加认证

修改配置文件

```
#vi /opt/logstash-1.4.2/conf/nginx_log.conf
```

```
output {  
  stdout { codec => rubydebug }  
  elasticsearch_http {  
    host => "192.168.0.100"  
    user => "es_admin"      #用户名  
    password => "123456"   #密码  
    port => "9200"  
  }  
}
```

## Elasticsearch 插件安装

Elasticsearch-head

功能: 集群管理工具

安装

```
#wget https://github.com/mobz/elasticsearch-head/archive/master.zip
```

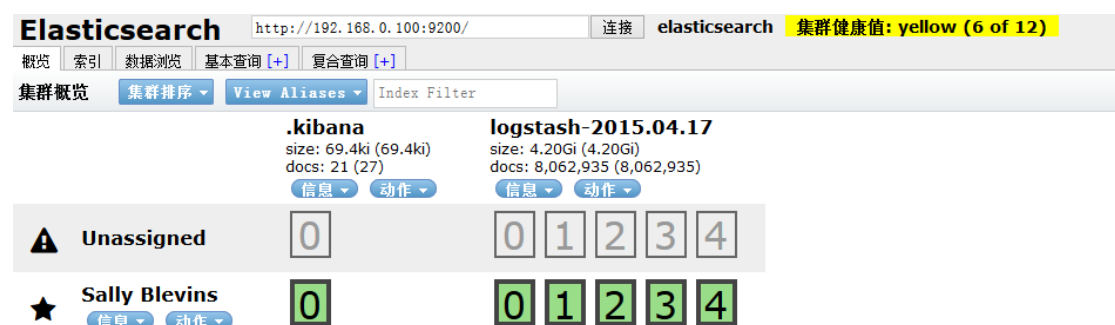
```
#unzip master.zip
```

```
#mkdir -p /opt/logstash/elasticsearch-1.4.4/plugins/head/_site
```

```
#mv elasticsearch-head-master/* /opt/logstash/elasticsearch-1.4.4/plugins/head/_site
```

访问 <http://192.168.0.100:9200/plugin/head/>

可方便查询索引数据及索引大小，如下图所示：





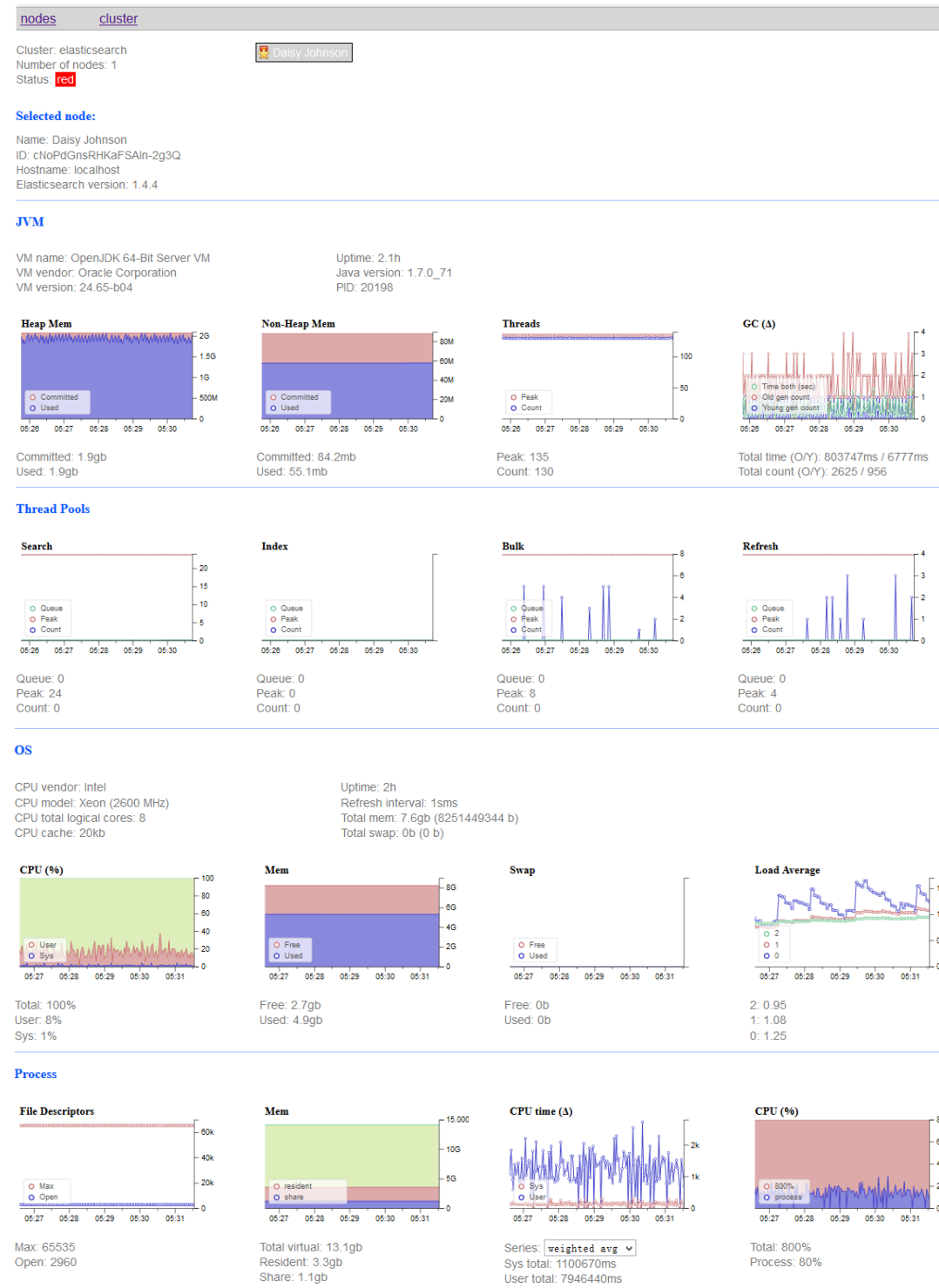
## Elasticsearch-bigdesk

**功能:** 监控查看 cpu、内存使用情况，索引数据、搜索情况，http 连接数等

### 安装

`#!/elstaicsearch/bin/plugin -i lukas-vlcek/bigdesk`

访问 [http://192.168.0.100:9200/\\_plugin/bigdesk](http://192.168.0.100:9200/_plugin/bigdesk)



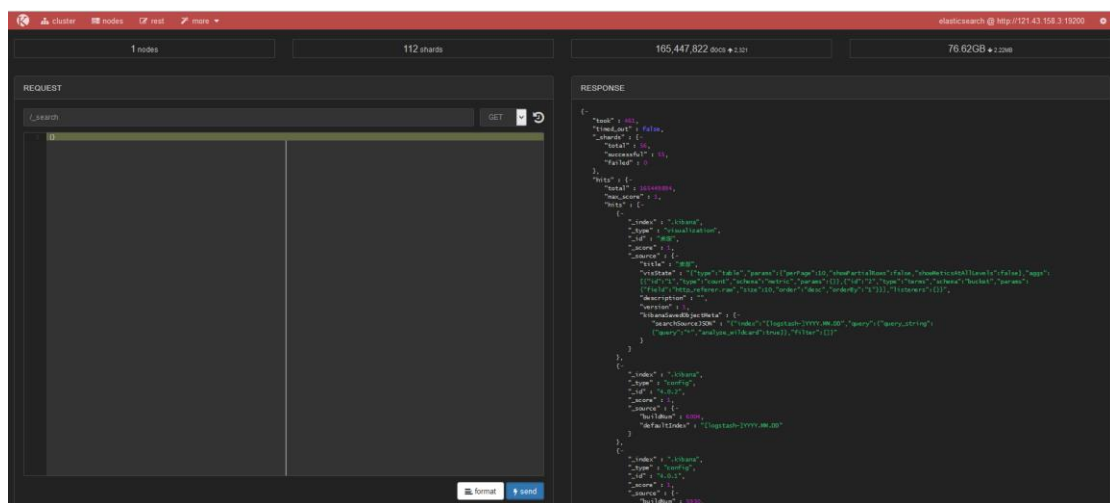
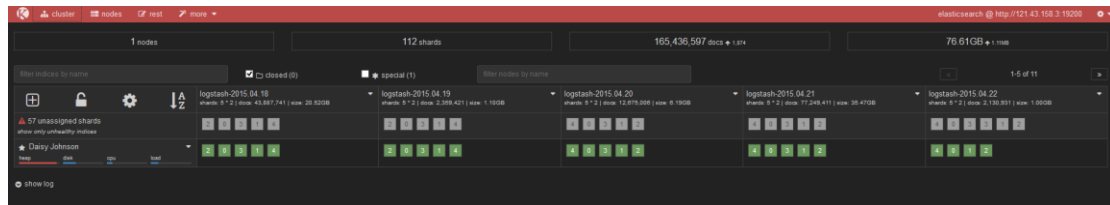
## Elasticsearch-kopf

**功能:** 集群管理, 监控查看 cpu、内存使用情况、在线查询/提交 elasticsearch 数据

## 安装

```
#/elasticsearch/bin/plugin -I lmenezes/elasticsearch-kopf
```

访问 [http://192.168.0.100:9200/\\_plugin/kopf](http://192.168.0.100:9200/_plugin/kopf)



## Elasticsearch 创建索引模板

默认调用的索引模板会自动分词，对数据分析不太方便。通过 curl 往 elasticsearch 里 PUT syslog-\* 索引模板。

```
# curl -ues_admin:123456 -XPUT http://192.168.0.100:9200/_template/template_syslog -d '
```

```
{
  "template" : "syslog-*",
  "settings" : {
    "index.refresh_interval" : "5s"
  },
  "mappings" : {
    "_default_" : {
      "_all" : {"enabled" : true},
      "dynamic_templates" : [ {
        "string_fields" : {
          "match" : "*",
          "match_mapping_type" : "string",
          "mapping" : {
            "type" : "string", "index" : "analyzed", "omit_norms" : true,
            "fields" : {
              "raw" : { "type": "string", "index" : "not_analyzed", "ignore_above" : 256,
"doc_values": true }
            }
          }
        }
      }
    ],
    "properties" : {
      "@version" : { "type": "string", "index": "not_analyzed" },
      "@timestamp" : { "type": "date", "index": "not_analyzed", "doc_values": true, "format":
"dateOptionalTime" },
      "geoip" : {
        "type" : "object",
        "dynamic": true,
        "path": "full",
        "properties" : {
          "location" : { "type" : "geo_point" }
        }
      }
    }
  }
}
```

注:

```
-ues_admin:123456          #elsticserach 用户名密码  
"type": "date", "index": "not_analyzed"  #索引不分词
```

在 logash 配置文件里就可以创建 syslog-开头的索引。Elasticsearch 会自动匹配 syslog-\*索引模板。

```
output {  
  stdout { codec => rubydebug }  
  elasticsearch_http {  
    host => "192.168.0.100"  
    user => "es_admin"  
    password => "123456"  
    port => "9200"  
    index => "syslog-%{+YYYY.MM.dd}"  
  }  
}
```

## Elasticsearch 常用操作

查询索引

```
#curl http://192.168.0.100:9200/syslog
```

```
#curl http://192.168.0.100:9200/\_aliases          #查询 elasticsearch 所有索引列表
```

创建,更新索引

```
# curl -XPUT http://192.168.0.100:9200/syslog -d '.....'
```

删除索引

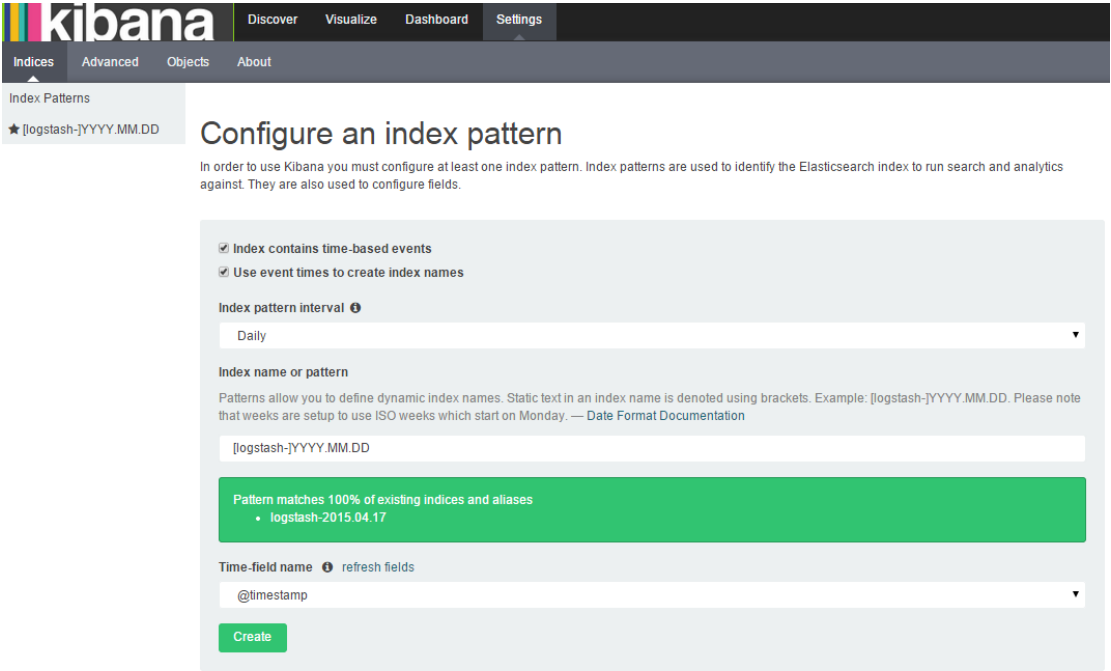
```
# curl -XDELETE http://192.168.0.100:9200/syslog
```

注:也可使用正则, #curl -XDELETE [http://192.168.0.100:9200/syslog\\*](http://192.168.0.100:9200/syslog*)

# Kibana 日志分析

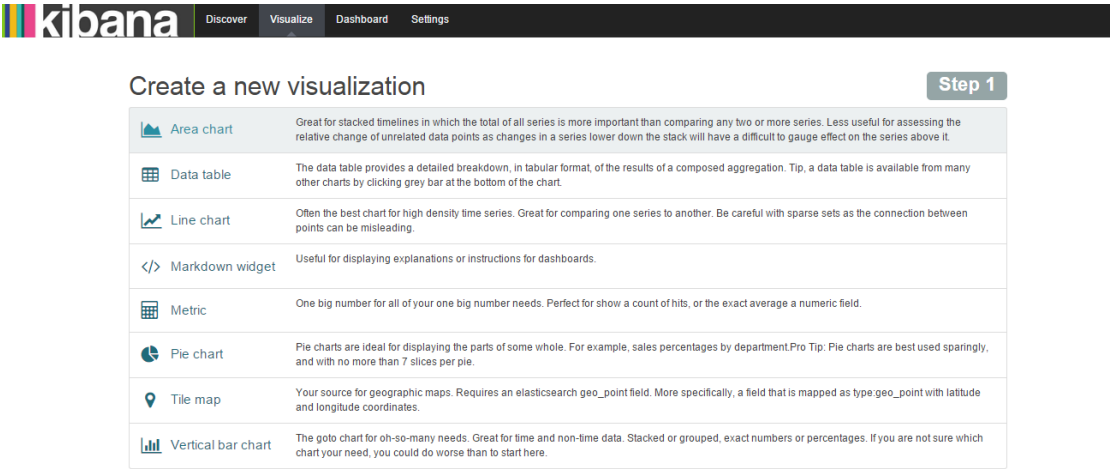
## 添加索引

访问 <http://192.168.0.100:5601>

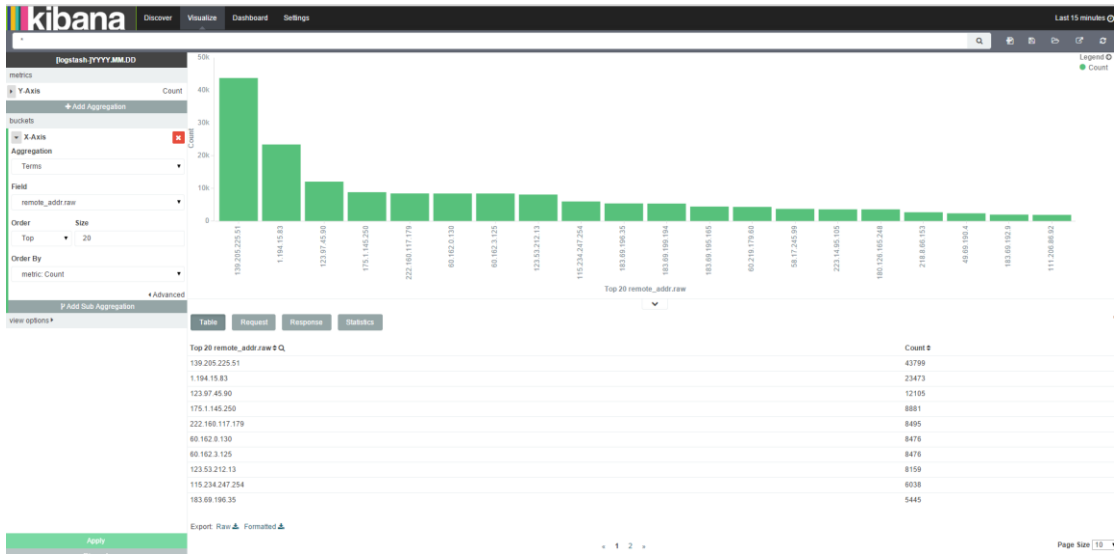


## 添加 Visualize 图表

有多种图表选择，如下图所示：



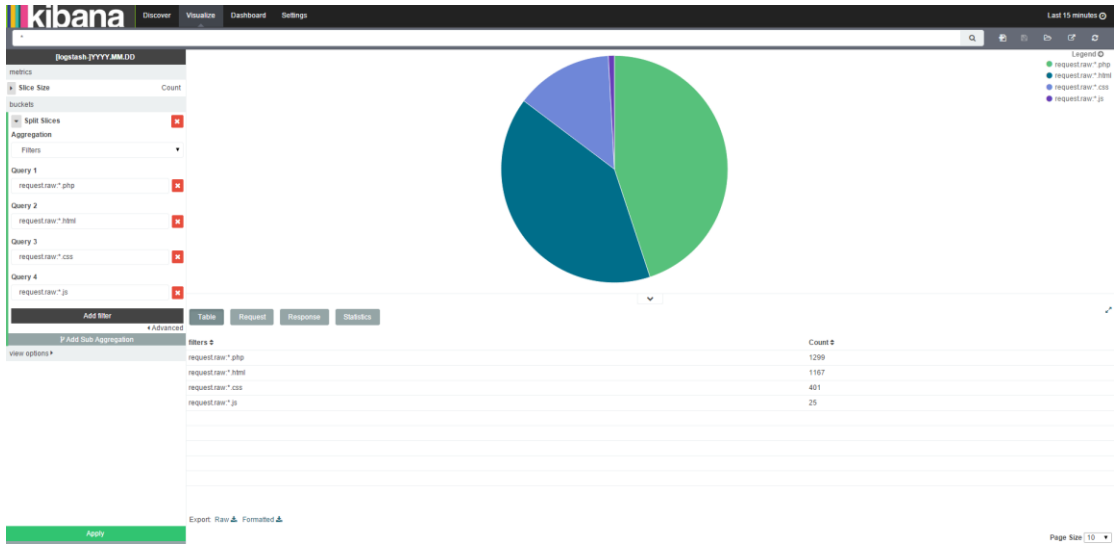
添加 Vertical bar chart 图表，统计访问前 20 的 IP



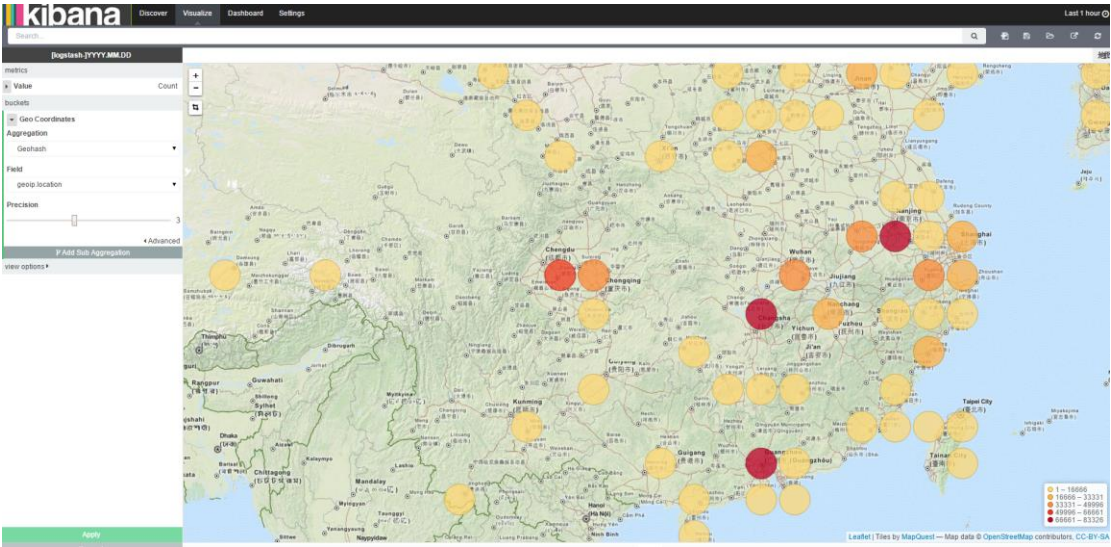
Metrics-Y-Axis 选择 Count  
Buckets-X-Axis-Aggregation 选择 Terms  
Field 选择 remote\_addr.raw

添加 Pie chart 图表

统计文件类型

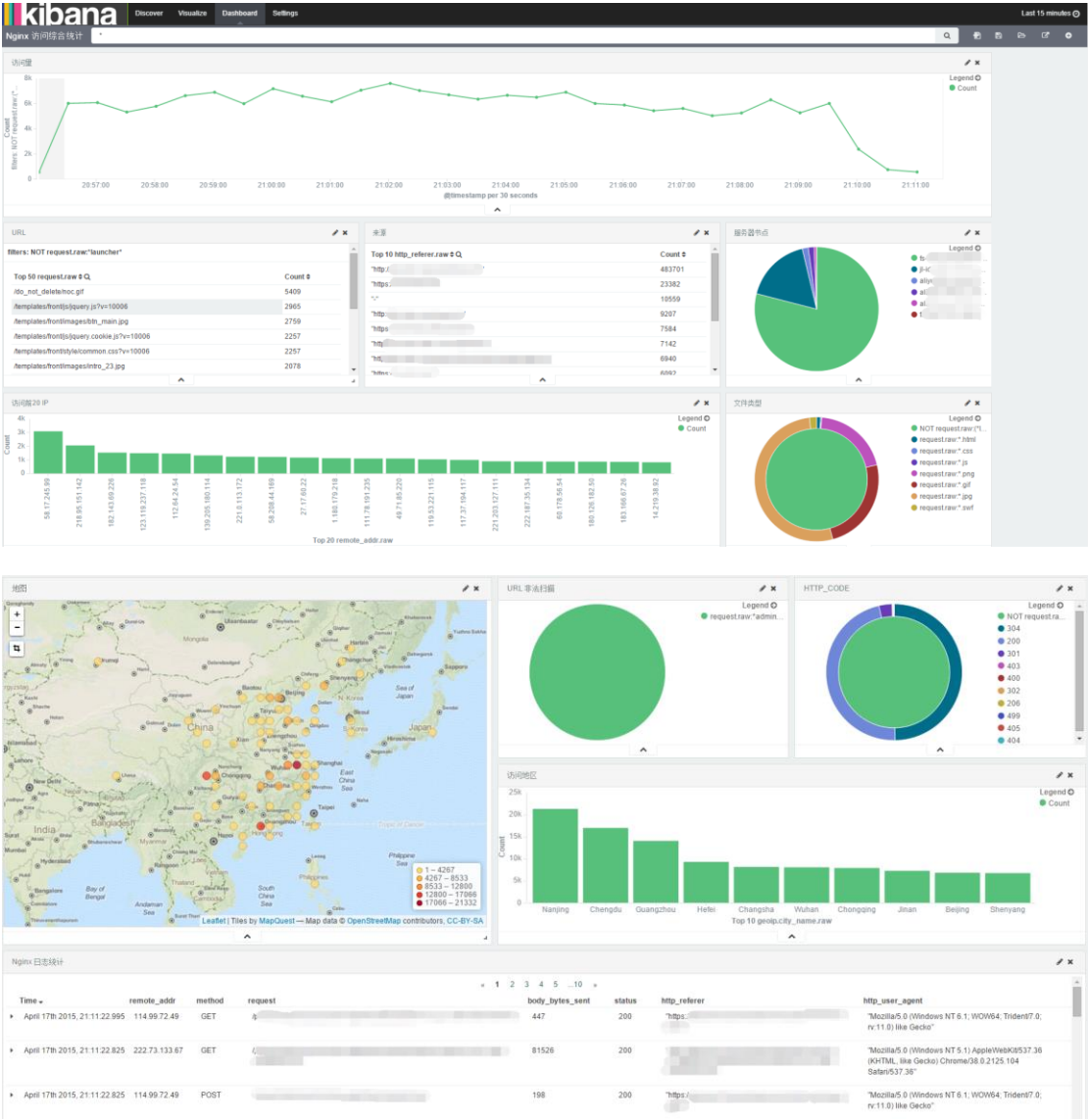


添加 Tile map 图表



# 添加 dashboard 图表

可以将多张 Visualize 图表添加到 dashboard 里。如下图所示:





# Kibana 统计网络带宽

由于 kibana sum 函数要求被统计的字段为 number 类型，默认 nginx 日志所有字段为 string 类型。

实例：统计 Nginx 占用带宽最高的前 10 条 URL

将以下内容添加到 logstash 配置文件的 `filter{}` 模块中

```
mutate {
  convert => ["body_bytes_sent","float"]    #将 body_bytes_sent 字段类型转换为 float 类型。
}
```

注:需要重新生成索引才能生效。

在 kibana 里添加图表

如下图所示：



# Kibana 查询说明

可使用 AND、OR、NOT 过滤字段条件。

**注:** AND、OR、NOT 必须大写

如:查询 request.raw 字段包含.rar 文件, 并且 status 不等于 200

request.raw:\*.rar AND NOT status.raw:200

如:查询 request.raw 字段包含.jsp 和 admin.php 或者 remote\_addr.raw 字段里包含 192.168 的 IP 地址

Request.raw:(\*.jsp \*admin.php) OR remote\_addr.raw:\*192.168

更多查询功能, 如下图所示:

## 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's 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.

### Refine your query

The search bar at the top uses Elasticsearch's support for Lucene Query String syntax. Let's say we're searching web server logs that have been parsed into a few fields.

Examples:

Find requests that contain the number 200, in any field:

200

Or we can search in a specific field. Find 200 in the status field:

status:200

Find all status codes between 400-499:

status:[400 TO 499]

Find status codes 400-499 with the extension php:

status:[400 TO 499] AND extension:PHP

Or HTML

status:[400 TO 499] AND (extension:php OR extension:html)