**logstash+kibana+elasticsearch+**

**redis搭建日志分析平台**

目录

[1目的 1](#_Toc467246507)

[2 ELK的介绍 3](#_Toc467246508)

[3架构描述 7](#_Toc467246509)

[4配置方法 8](#_Toc467246510)

[5使用 18](#_Toc467246511)

[6常见问题 30](#_Toc467246512)

# 1目的

日志的分析在系统开发中占非常重要的地位，系统越复杂，日志的分析就越重要，常见的需求有:

• 根据关键字查询日志详情

• 基于日志的数据挖掘

很多团队在日志方面可能遇到的一些问题有:

• 开发人员不能登录线上服务器查看详细日志，经过运维周转费时费力

• 日志数据分散在多个系统，难以查找

• 日志数据量大，查询速度慢

• 一个调用会涉及多个系统，难以在这些系统的日志中快速定位数据

• 数据不够实时

常见的一些重量级的开源Trace系统有

• facebook scribe

• Apache Chukwa

• Linkedin Kafak

• Cloudera flumentd

• twitter zipkin

• storm

还有简单的 Rsyslog，Syslog-ng；商业化的 Splunk ；在上述产品中，Splunk 是一款非常优秀的产品，但是它是商业产品，价格昂贵，让许多人望而却步。而其他项目功能也很强大，但对于很多团队来说过于复杂，配置和部署比较麻烦，在系统规模大到一定程度前一般都推荐轻量级下载即用的方案

直到 ELK 的出现，让大家又多了一种选择。相对于其他几款开源软件来说，本文重点介绍 ELK。

即elasticsearch+logstash+kibana(ELK)组合。

对于日志来说，最常见的需求就是收集、查询、显示，正对应logstash、elasticsearch、kibana的功能。

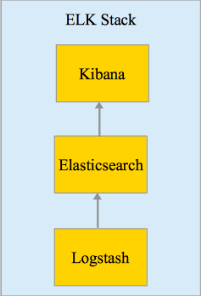
Logstash是一个完全开源的工具，他可以对你的日志进行收集、分析，并将其存储供以后使用（如，搜索），您可以使用它。说到搜索，logstash带有一个web界面，搜索和展示所有日志。  
kibana 也是一个开源和免费的工具，他可以帮助您汇总、分析和搜索重要数据日志并提供友好的web界面。他可以为 Logstash 和 ElasticSearch 提供的日志分析的 Web 界面

# 2 ELK的介绍

ELK 协议栈介绍及体系结构

ELK 其实并不是一款软件，而是一整套解决方案，是三个软件产品的首字母缩写，Elasticsearch，Logstash 和 Kibana。这三款软件都是开源软件，通常是配合使用，而且又先后归于 Elastic.co 公司名下，故被简称为 ELK 协议栈，见图 1。

图 1.ELK 协议栈



Elasticsearch

Elasticsearch 是一个实时的分布式搜索和分析引擎，它可以用于全文搜索，结构化搜索以及分析。它是一个建立在全文搜索引擎 Apache Lucene 基础上的搜索引擎，使用 Java 语言编写。目前，最新的版本是 2.1.0。

主要特点

实时分析

分布式实时文件存储，并将每一个字段都编入索引

文档导向，所有的对象全部是文档

高可用性，易扩展，支持集群（Cluster）、分片和复制（Shards 和 Replicas）。见图 2 和图 3

接口友好，支持 JSON

图 2. 集群

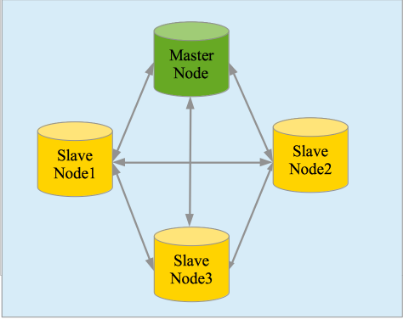
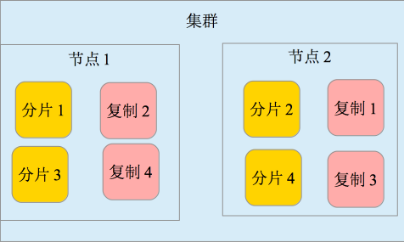


图 3. 分片和复制



Logstash

Logstash 是一个具有实时渠道能力的数据收集引擎。使用 JRuby 语言编写。其作者是世界著名的运维工程师乔丹西塞 (JordanSissel)。目前最新的版本是 2.1.1。

主要特点

几乎可以访问任何数据

可以和多种外部应用结合

支持弹性扩展

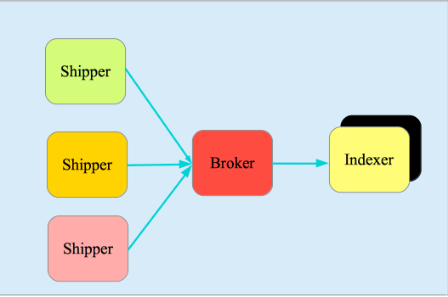
它由三个主要部分组成，见图 4：

Shipper－发送日志数据

Broker－收集数据，缺省内置 Redis

Indexer－数据写入

图 4.Logstash 基本组成



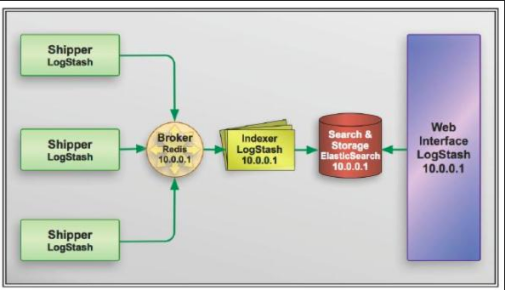
Kibana

Kibana 是一款基于 Apache 开源协议，使用 JavaScript 语言编写，为 Elasticsearch 提供分析和可视化的 Web 平台。它可以在 Elasticsearch 的索引中查找，交互数据，并生成各种维度的表图。目前最新的版本是 4.3，简称 Kibana 4。

ELK 协议栈体系结构

完整的 ELK 协议栈体系结构见图 5。基本流程是 Shipper 负责从各种数据源里采集数据，然后发送到 Broker，Indexer 将存放在 Broker 中的数据再写入 Elasticsearch，Elasticsearch 对这些数据创建索引，然后由 Kibana 对其进行各种分析并以图表的形式展示。

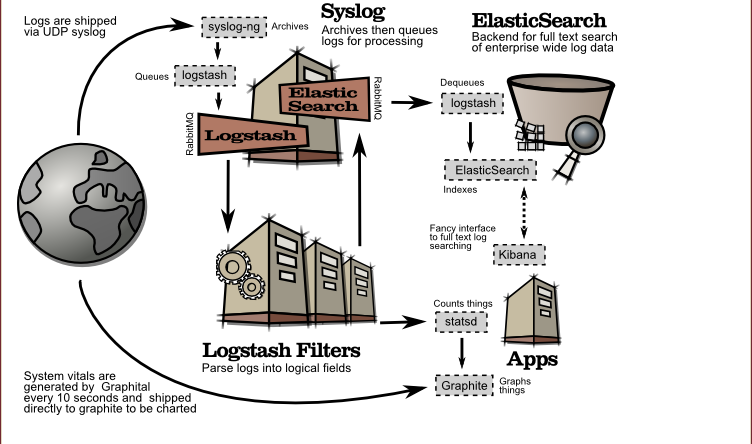
图 5.ELK 协议栈体系结构

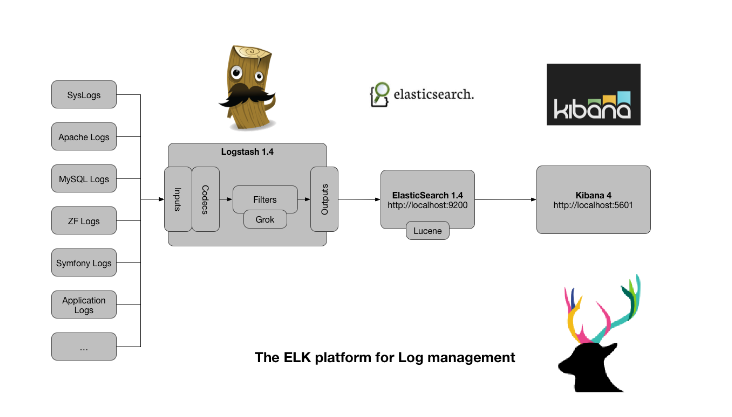


ELK 三款软件之间互相配合使用，完美衔接，高效的满足了很多场合的应用，并且被很多用户所采纳，诸如路透社，脸书（Facebook），StackOverFlow 等等。

# 3架构描述

简单来讲他具体的工作流程就是 logstash agent 监控并过滤日志，将过滤后的日志内容发给redis(这里的redis只处理队列不做存储)，logstash index将日志收集在一起交给全文搜索服务ElasticSearch 可以用ElasticSearch进行自定义搜索 通过Kibana 来结合 自定义搜索进行页面展示，下图是 Kibana官网上的流程图





上图是该套系统的一个工作流程。

Logstash：负责日志的收集，处理和储存。

Elasticsearch：负责日志检索和分析。

Kibana：负责日志的可视化。

# 4配置方法

好了 让我们一步步的把这套环境搭建起来吧，先看看都需要安装什么软件包

ruby 运行Kibana 必须，

rubygems 安装ruby扩展必须

bundler 功能类似于yum

JDK 运行java程序必须

redis 用来处理日志队列

logstash 收集、过滤日志

ElasticSearch 全文搜索服务

kibana 页面展示

这里有四台服务器

192.168.8.15 kibana,JDK

192.168.8.17 logstash index,JDK

192.168.8.19 Redis,JDK

192.168.8.21 ElasticSearch,JDK

当然只要服务器资源足够，理论上这些组件都部署在一台服务器上也是可以的。

**首先在8.19这台机器上安装redis:**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **IP** | **数据目录** | **端口** |
| Master | 192.168.8.19 | 日志：  /var/log/redis/redis.log  配置文件：  /usr/local/bin  数据文件：  /etc/redis | 6379 |

#cd /root

#mkdir redis

拷贝redis-stable.tar.gz到/root/redis文件夹

#mkdir /etc/redis

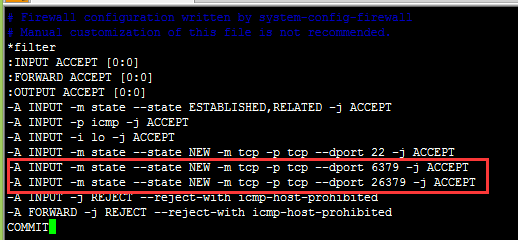
创建数据目录

#mkdir /var/log/redis

创建日志目录

#vim /etc/sysconfig/iptables

打开防火墙6379端口



# service iptables restart

重启防火墙

#yum install -y gcc-c++

安装gcc编译器

#cd /root/redis

# tar xvzf redis-stable.tar.gz

#cd redis-stable

#make

#yum install -y tcl

#make test

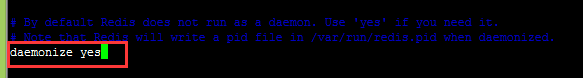
#make install

#cp redis.conf /usr/local/bin

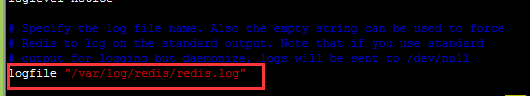
#cd /usr/local/bin

#vim /usr/local/bin/redis.conf

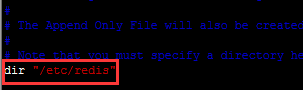
修改daemonize为yes



修改logfile



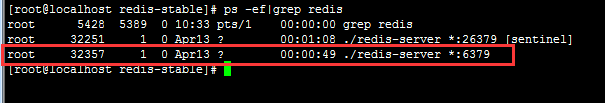
修改dir



#./redis-server redis.conf

**启动redis**

# ps -ef|grep redis



温馨提示：为了方便，最好简单粗暴把防火墙关掉

（# service iptables stop）

**然后在8.21这台机器上边安装Elasticsearch:**

#cd /home

#mkdir elsearch

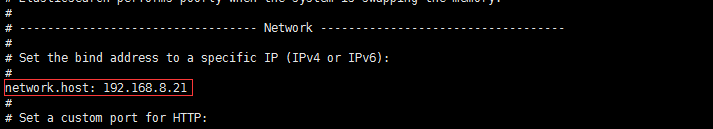
拷贝elasticsearch-2.3.3.tar.gz到/home/elsearch文件夹

# tar xvzf elasticsearch-2.3.3.tar.gz

# vim elasticsearch.yml

(该文件在/home/elsearch/elasticsearch-2.3.3/config文件夹下)

将此处修改为本机ip



(此处声明：必须用其他用户启动，root用户启动会报错)

所以先创建elsearch用户组及elsearch用户

#groupadd elsearch

#useradd elsearch -g elsearch -p elsearch

更改elsearch文件夹及内部文件的所属用户及组为elsearch:elsearch

#chown -R elsearch:elsearch elsearch

启动elasticsearch

#/home/elsearch/elasticsearch-2.3.3/bin/elasticsearch -d

**然后在8.17这台机器上装logstash:**

logstash分为 index和agent ，agent负责监控、过滤日志，index负责收集日志并将日志交给ElasticSearch 做搜索

此外 logstash 的收集方式分为 standalone 和 centralized。

standalone 是所有功能都在一个服务器上面，自发自收，centralized 就是集中收集，一台服务器接收所有shipper(个人理解就是logstash agent)的日志。

其实 logstash本身不分 什么 shipper 和 collector ，只不过就是配置文件不同而已。

#cd /root

#mkdir soft

拷贝logstash-all-plugins-2.3.1.zip到/root/soft文件夹

#unzip logstash-all-plugins-2.3.1.zip

增加/修改redis.conf配置文件

#vim redis.conf

**input {**

**stdin {**

**}**

**redis {**

**host => "192.168.8.19"** #这里定义redis主机地址

**data\_type => "list"**

**port => "6379"**

**key => "logstash:redis"**

**type => "redis-input"**

**codec => "json"**

**}**

**}**

**output {**

**elasticsearch {**

**hosts => ["192.168.8.21:9200"]** #这里定义elasticsearch主机地址和端口

**}**

**}**

解释一下 logstash的配置文件由 input filter output 等几个基本的部分组成，顾名思义 input 就是在那收集数据，output就是输出到哪，filter代表一个过滤规则意思是什么内容

会被收集。

上面这段是让 logstash 去192.168.8.19 这个redis服务器上去收集日志 redis端口为6379，key是 logstash:redis 类型为 redis-input ，（注意:这几个值必须跟logstash agent的

output 所对应），收集完成后输出到 elasticsearch ,embedded => true 的意思是使用logstash 内嵌的 elasticsearch。如果有独立的elasticsearch服务器，需要将 这条改为

host => 'elasticsearch的ip' port => 端口

好了，这个简单的配置文件可以让logstash开始启动了

启动logstash

#nohup /root/soft/logstash-2.3.1/bin/logstash –f /root/soft/redis.conf &

**还有我们也同样需要在需要被分析日志的服务器上装logstash：**

#cd /root

#mkdir soft

拷贝logstash-all-plugins-2.3.1.zip到/root/soft文件夹

#unzip logstash-all-plugins-2.3.1.zip

增加/修改redis.conf配置文件

#vim redis.conf

**input {**

**file {**

**type => "localhostlog"** #这里定义的是日志文件名

**path => ["/root/mail/nohup.out"]** #这里定义的是日志文件路径

**}**

**}**

**output {**

**redis {**

**host => "192.168.8.19"** #这里定义redis主机地址

**data\_type => "list"**

**key => "logstash:redis"**

**}**

**}**

大概说一下这个配置文件 input 里的file就是要监视的文件了，如果这个文件有追加的内容就会通过下面的output设置发给 redis服务器

启动logstash

#nohup /root/soft/logstash-2.3.1/bin/logstash –f /root/soft/redis.conf &

**最后在8.15这台机器上安装kibana:**

安装ruby 就比较简单了(Kibana需要ruby 1.8.7以上版本)

#yum install ruby rubygems

..... 安装内容省略

安装完成后用 rubygems 来安装bundler

#/usr/bin/gem install bundler

#cd /root

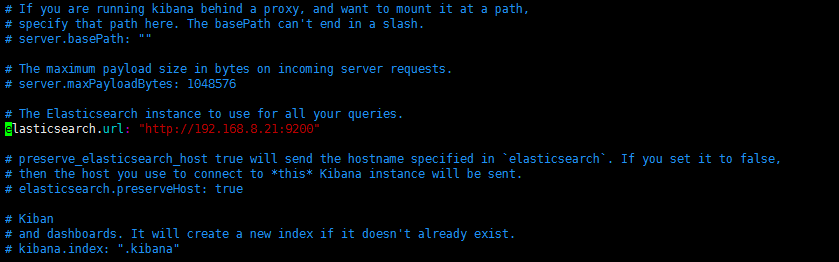
#mkdir soft

拷贝kibana-4.5.1-linux-x64.tar.gz到/root/soft文件夹

# tar xvzf kibana-4.5.1-linux-x64.tar.gz

修改kibana.yml文件中，elasticsearch URL地址

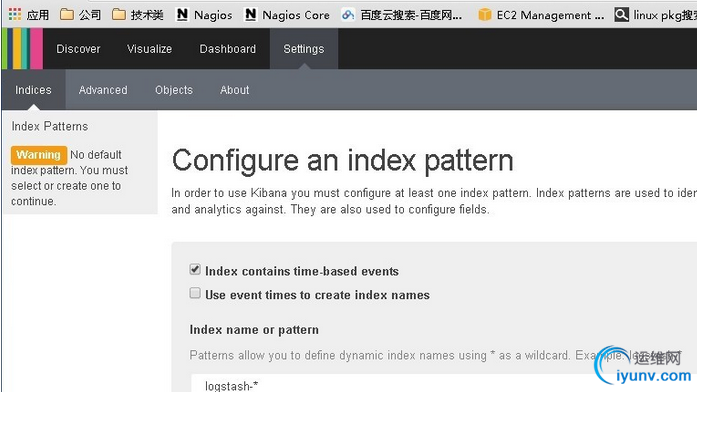
#vim /root/soft/kibana-4.5.1-linux-x64/config/ kibana.yml



启动kibana

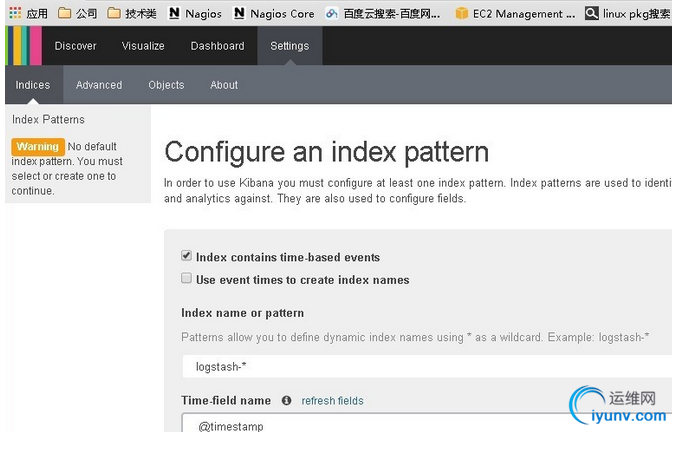
#nohup /root/soft/kibana-4.5.1-linux-x64/bin/kibnan &

这样这套系统就搭起来了，浏览器输入：192.168.8.15:5601看到如下页：

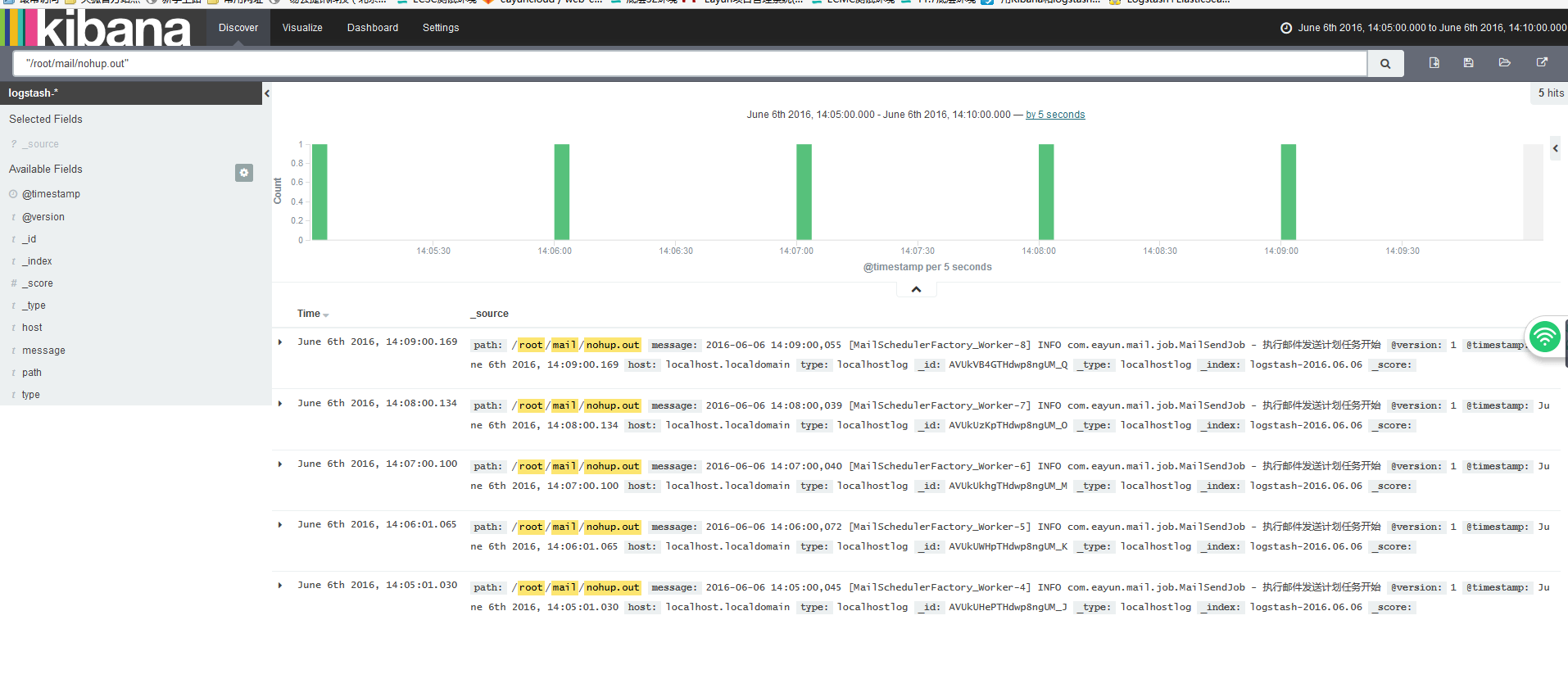


进来之后他要你创建一个默认索引，可是怎么办没有create啊，下面会灰色的，是不是我做错了。好吧，其实都不是，这个如果你装完kibana你那个日志文件要是一直没有新的日志产生呢这里就一直是灰的，这个时候你只需要去访问一下你的网站，然后就行了。

然后你就可以看到下面的情况了，选择timestamp，点击create创建，完成。

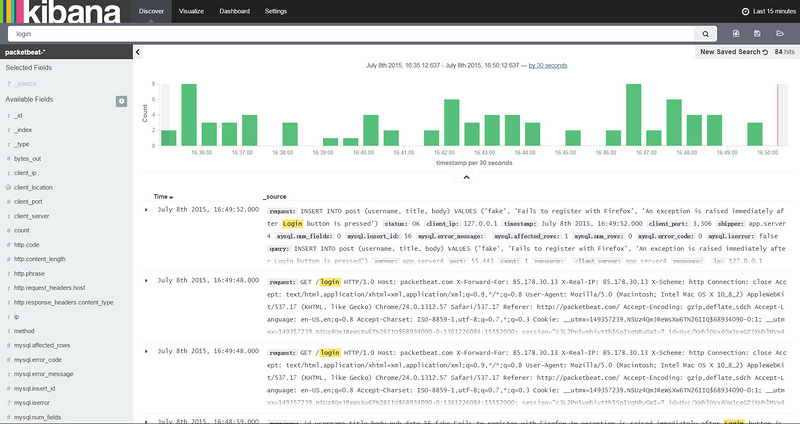


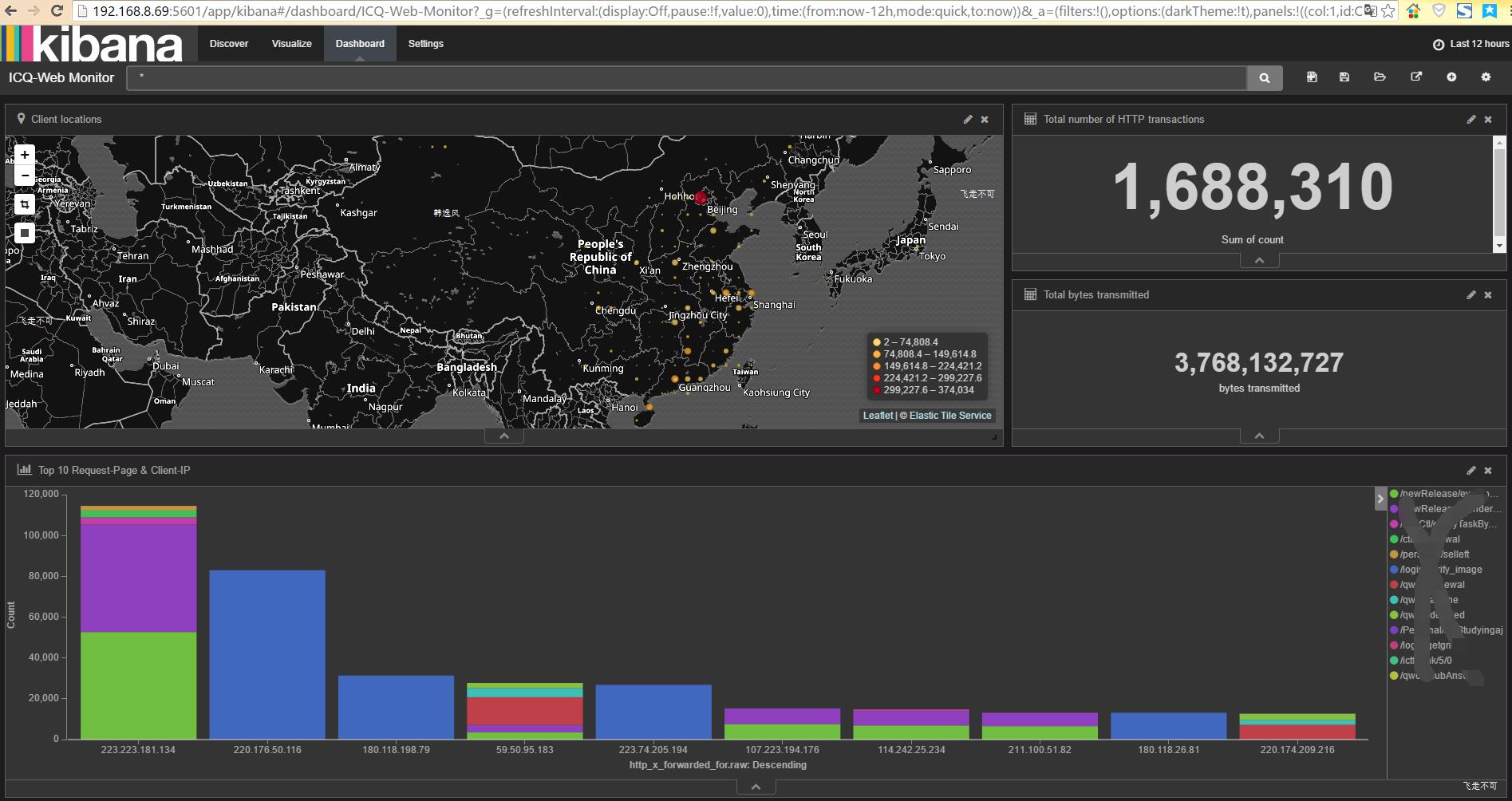
搭建成功

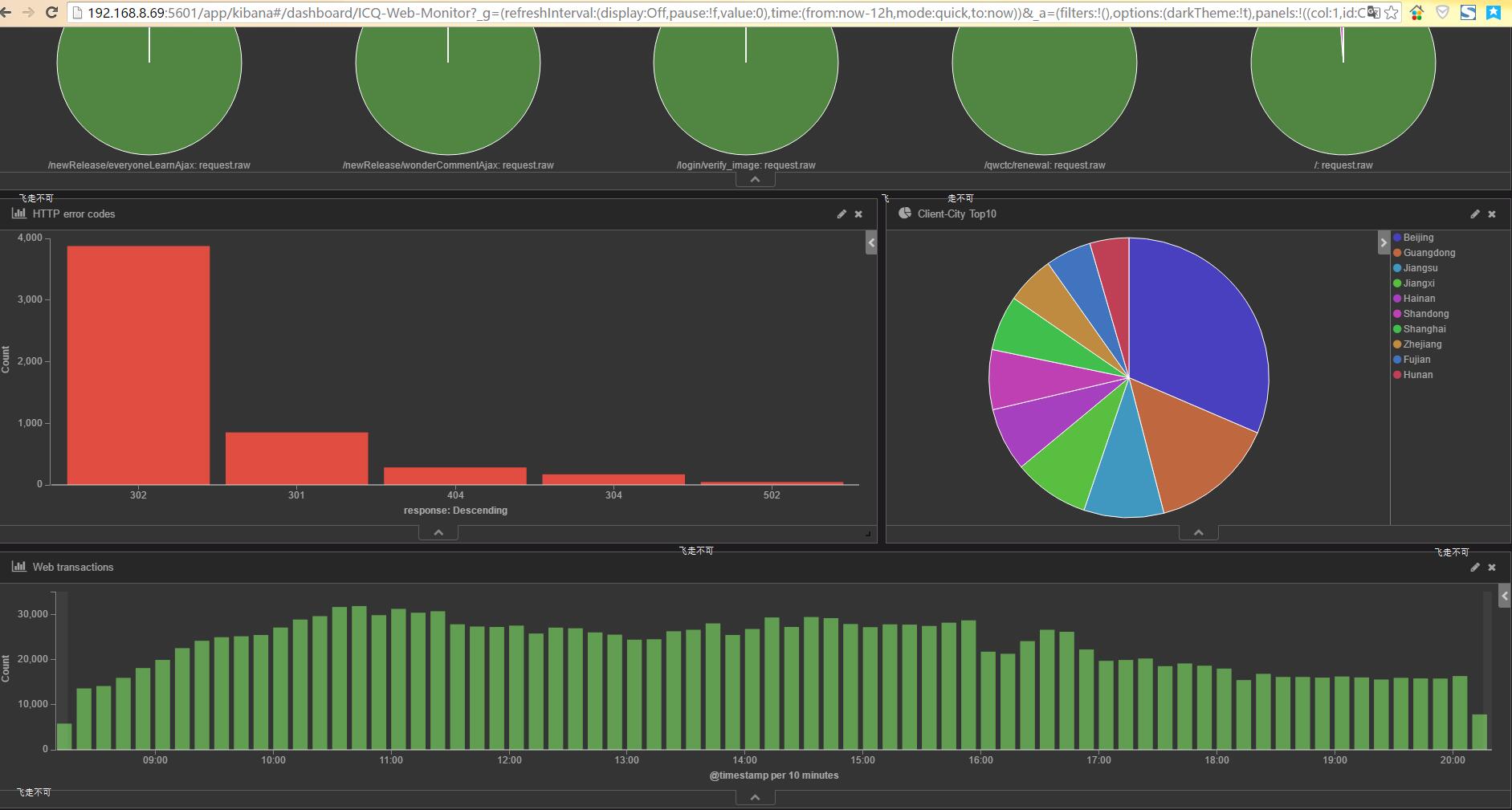


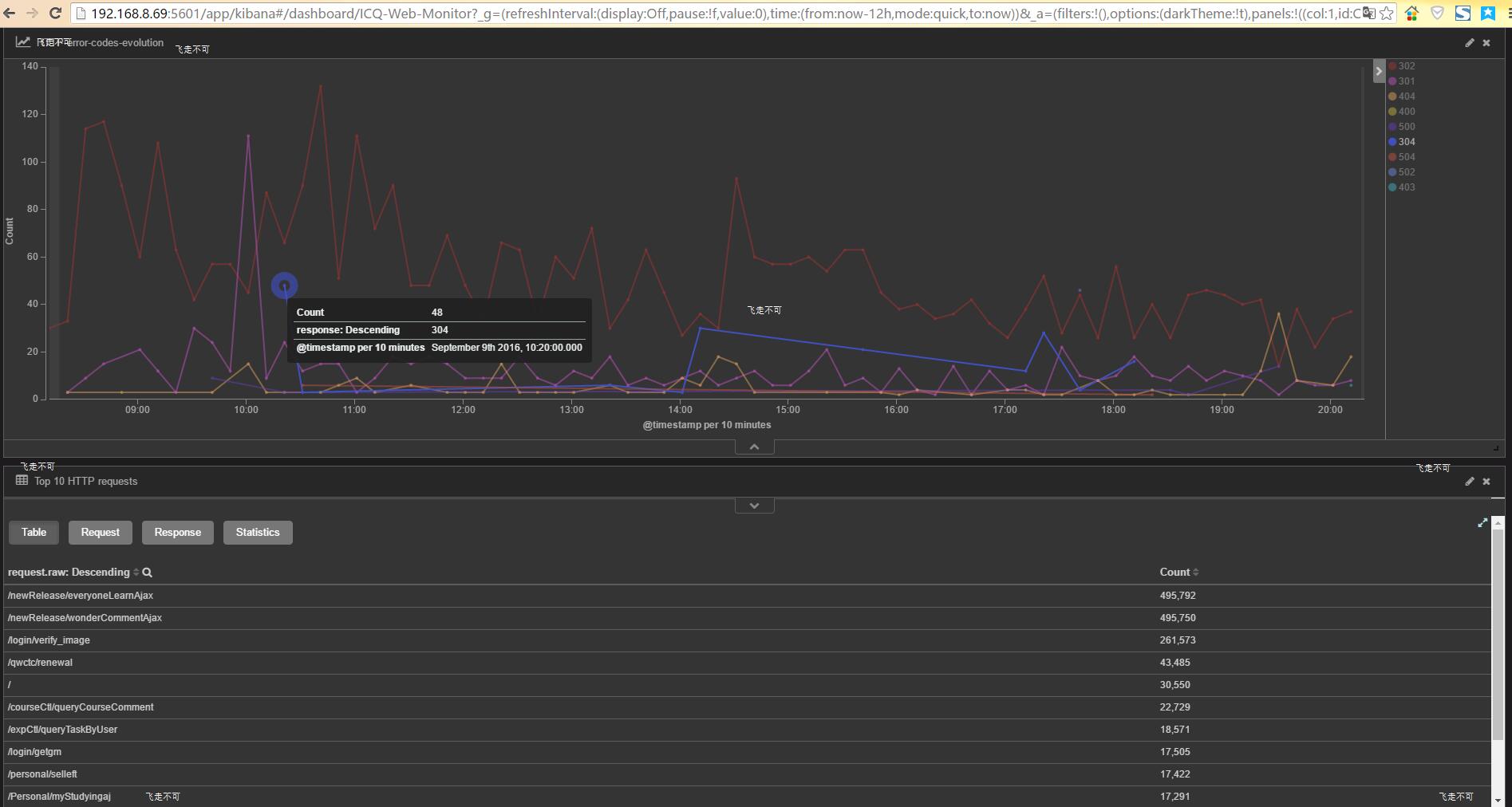
# 5使用

**全文搜索**

在搜索栏输入login，会返回所有字段值中包含login的文档  








上面图主要统计了web站点的客户端ip地址归属地、总的http传输次数、top10 来源ip、top10 请求点击页面、错误请求趋势、等等。

使用双引号包起来作为一个短语搜索  
"like Gecko"

**字段**

也可以按页面左侧显示的字段搜索  
限定字段全文搜索：field:value  
精确搜索：关键字加上双引号 filed:"value"  
http.code:404 搜索http状态码为404的文档

字段本身是否存在  
\_exists\_:http：返回结果中需要有http字段  
\_missing\_:http：不能含有http字段

**通配符**

? 匹配单个字符  
\* 匹配0到多个字符

kiba?a, el\*search

? \* 不能用作第一个字符，例如：?text \*text

**正则**

es支持部分[正则](https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl-regexp-query.html" \l "regexp-syntax" \t "_blank)功能  
mesg:/mes{2}ages?/

**模糊搜索**

~:在一个单词后面加上~启用模糊搜索

first~ 也能匹配到 frist

还可以指定需要多少相似度  
cromm~0.3 会匹配到 from 和 chrome  
数值范围0.0 ~ 1.0，默认0.5，越大越接近搜索的原始值

**近似搜索**

在短语后面加上~  
"select where"~3 表示 select 和 where 中间隔着3个单词以内

**范围搜索**

数值和时间类型的字段可以对某一范围进行查询  
length:[100 TO 200]  
date:{"now-6h" TO "now"}  
[ ] 表示端点数值包含在范围内，{ } 表示端点数值不包含在范围内

**逻辑操作**

AND  
OR account\_number:<100 AND balance:>47500

+：搜索结果中必须包含此项  
-：不能含有此项  
+apache -jakarta test：结果中必须存在apache，不能有jakarta，test可有可无

**分组**

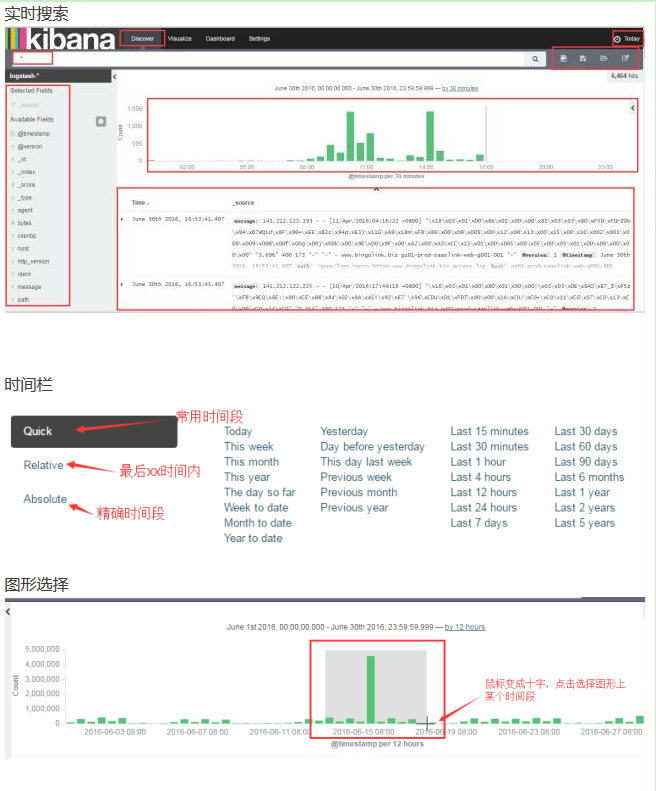
(jakarta OR apache) AND Jakarta

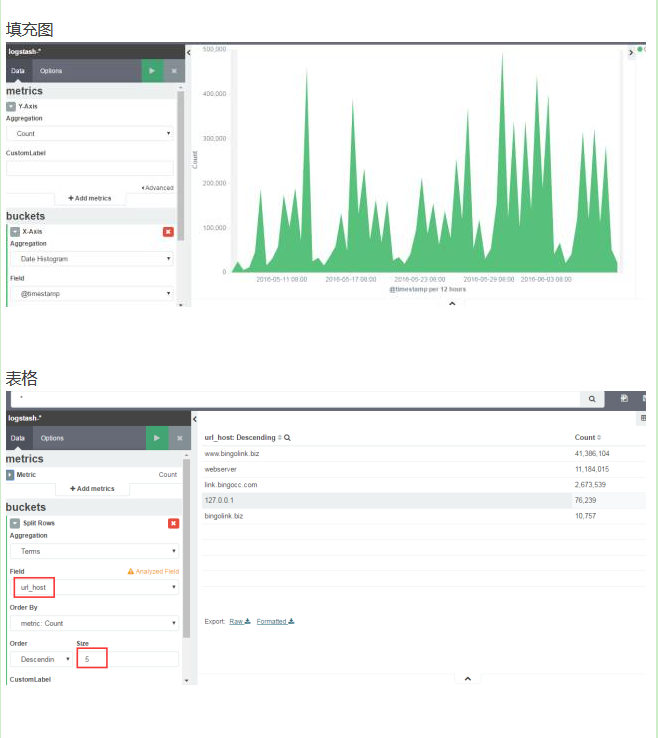
**字段分组**

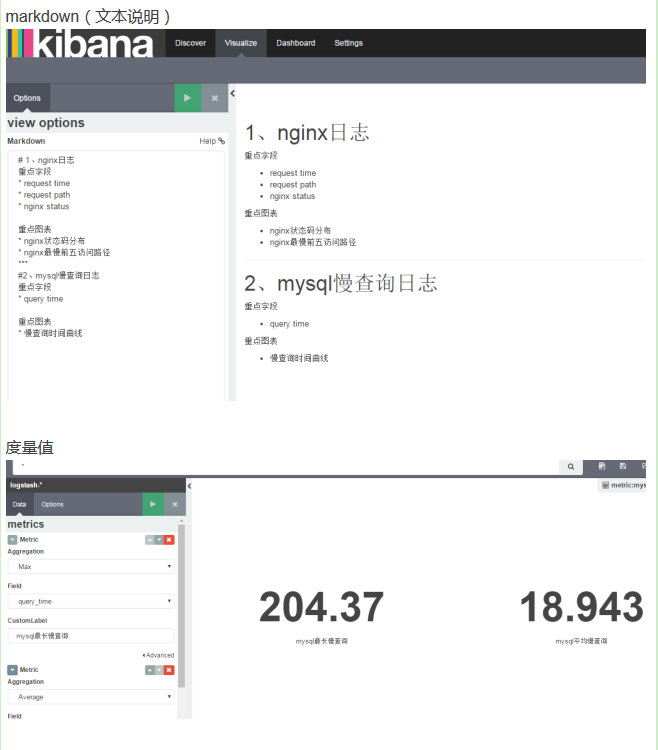
title:(+return +"pink panther")

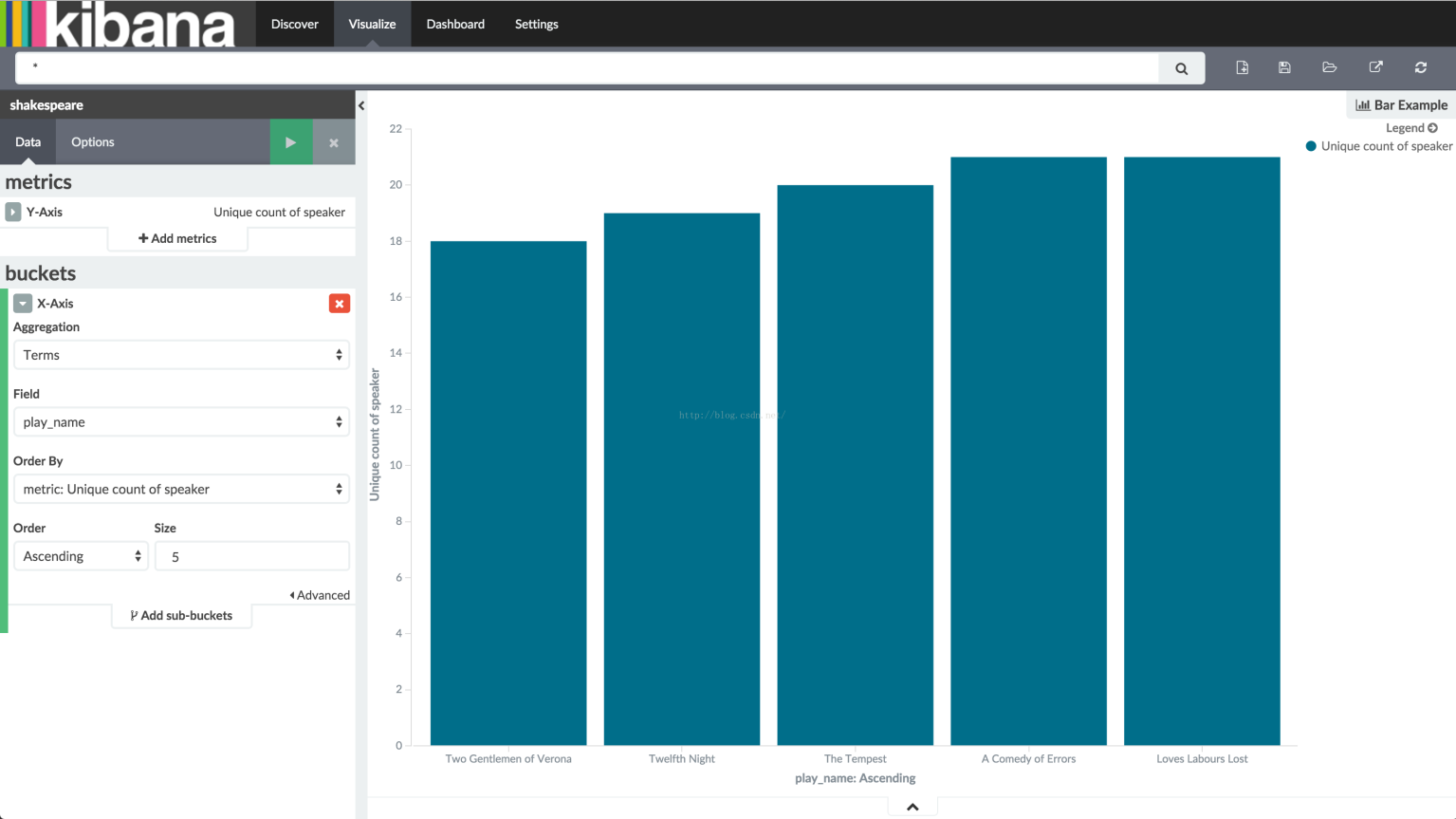
**转义特殊字符**

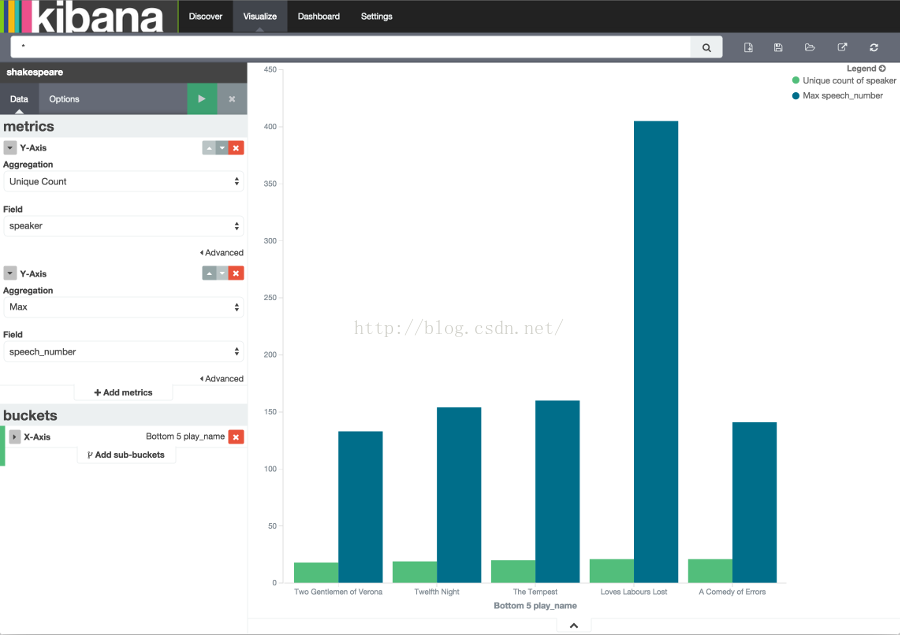
+ - && || ! () {} [] ^" ~ \* ? : \  
以上字符当作值搜索的时候需要用\转义

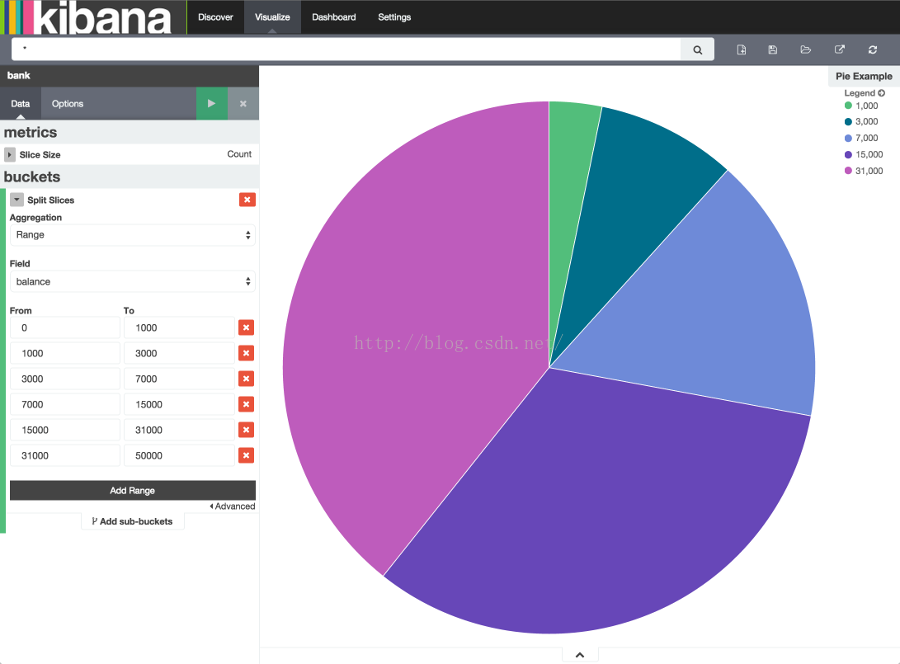


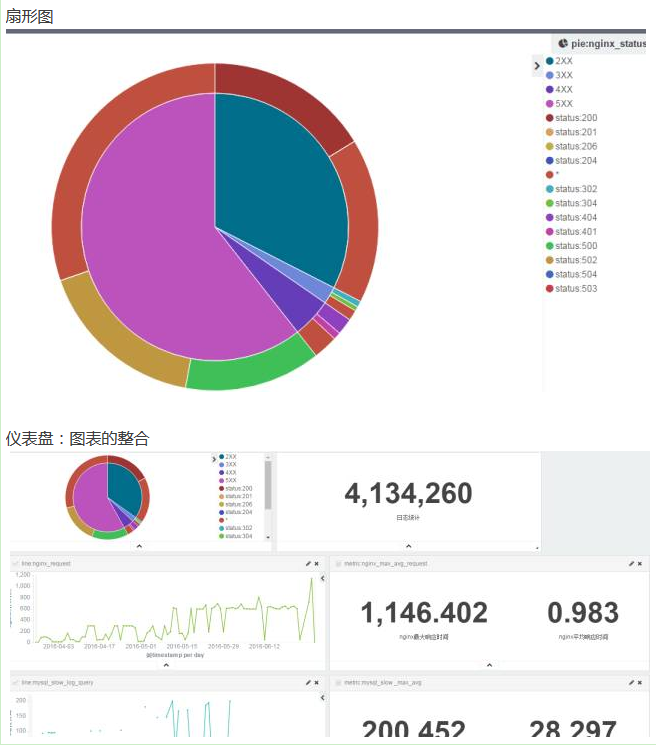












# 6常见问题

**logstash无法启动，提示bind address**

原因：

1）配置目录下存在多个配置文件，而logstash会加载所有conf格式的文件

解决方案：删除不必要的文件，保留一个conf文件即可

2）进程未结束

解决方案：kill -9 pid 强制结束进程，再启动服务即可

**字段无法解析 \_grokparsefailure**

kibana无法解析出相应的字段

原因：正则存在问题或者日志不符合正则格式

解决方案：在http://grokdebug.herokuapp.com/上调试正则，同时确保日志中不存在多余空格等异常

还有一种常见原因：空格、空格、空格，重要的事情说三遍！

**日志量大，磁盘紧张**

日志量增加非常快，磁盘空间不够用怎么办？可以通过删除较早的索引来缓解

path.data: /data/elasticsearch/data 数据目录存放位置

path.logs: /data/elasticsearch/log 日志数据存放位置