

大型架构及配置技术

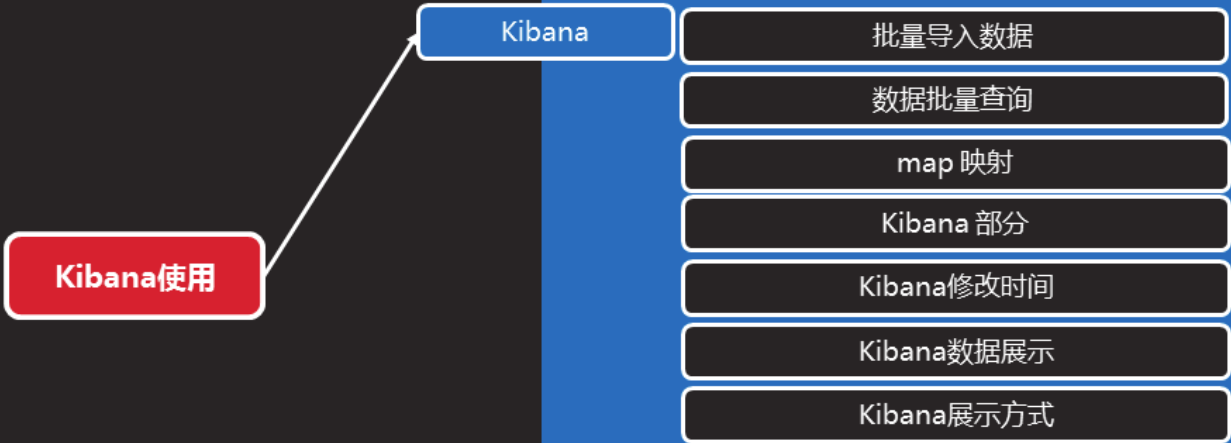
NSD ARCHITECTURE **DAY04**

内容

上午	09:00 ~ 09:30	作业讲解和回顾
	09:30 ~ 10:20	Kibana使用
	10:30 ~ 11:20	
	11:30 ~ 12:00	
下午	14:00 ~ 14:50	Logstash配置 扩展插件
	15:00 ~ 15:50	
	16:10 ~ 17:10	
	17:20 ~ 18:00	总结和答疑



Kibana使用



Kibana

批量导入数据

- 使用_bulk批量导入数据
 - 批量导入数据使用POST方式，数据格式为json，url编码使用data-binary
 - 导入含有index配置的json文件

```
# gzip -d logs.jsonl.gz
# curl -XPOST 'http://192.168.4.14:9200/_bulk' --data-binary @logs.jsonl

# gzip -d shakespeare.json.gz
# curl -XPOST 'http://192.168.4.14:9200/_bulk' --data-binary @shakespeare.json
```

批量导入数据（续1）

知识讲解

- 使用_bulk批量导入数据
 - 导入没有index配置的json文件
 - 我们需要在ur里面制定index和type

```
# gzip -d accounts.json.gz
# curl -XPOST
'http://192.168.4.14:9200/accounts/act/_bulk?pretty' --data-
binary @accounts.json
```



数据批量查询

知识讲解

- 数据批量查询使用GET

```
# curl -XGET 'http://192.168.4.11:9200/_mget?pretty' -d '{
  "docs":[
    {
      "_index": "accounts",
      "_type": "act",
      "_id": 1
    },
    {
      "_index": "accounts",
      "_type": "act",
      "_id": 2
    }
  ]
}
```



数据批量查询（续1）

- 数据批量查询使用GET

- 续上一页

```
{  
  "_index": "shakespeare",  
  "_type": "scene",  
  "_id": 1  
}  
]  
'
```

知识讲解



map 映射

- mapping :
 - 映射：创建索引的时候，可以预先定义字段的类型及相关属性
 - 作用：这样会让索引建立得更加细致和完善
 - 分类：静态映射和动态映射
 - 动态映射：自动根据数据进行相应的映射
 - 静态映射：自定义字段映射数据类型

知识讲解



案例1：导入数据

1. 批量导入数据并查看

课堂练习



Kibana 部分

- 数据导入以后查看logs是否导入成功

知识讲解

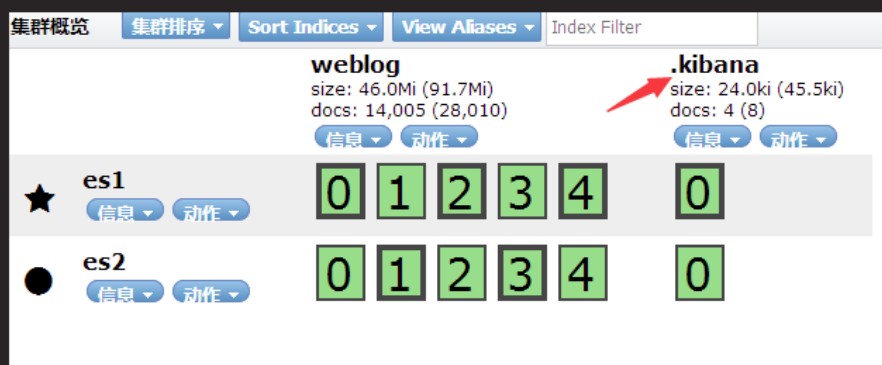
logstash-2015.05.20	logstash-2015.05.19	logstash-2015.05.18
size: 27.3Mi (57.7Mi) docs: 4,750 (9,500)	size: 25.4Mi (58.6Mi) docs: 4,624 (9,248)	size: 26.8Mi (58.5Mi) docs: 4,631 (9,262)
信息 动作	信息 动作	信息 动作
<div>0 1 2 3</div> <div>4</div>	<div>0 1 2 3</div> <div>4</div>	<div>0 1 2 3</div> <div>4</div>
<div>0 1 2 3</div> <div>4</div>	<div>0 1 2 3</div> <div>4</div>	<div>0 1 2 3</div> <div>4</div>



Kibana 部分（续1）

- 修改Kibana的配置文件后启动Kibana，然后查看ES集群，如果出现.kibana Index表示Kibana与ES集群连接成功

知识讲解



Kibana 部分（续2）

- Kibana里选择日志
 - 支持通配符 *
 - 我们这里选择logstash-*
 - 在下面的Time-field选择@timestamp作为索引
 - 然后点create按钮

知识讲解



Kibana 部分 (续3)

知识讲解

Configure an index pattern

In order to use Kibana you must configure at least one index pattern. Index patterns are used to identify the Elasticsearch index to run search and analytics against. They are also used to configure fields.

☒ Index contains time-based events

☐ Use event times to create index names [DEPRECATED]

Index name or pattern

Patterns allow you to define dynamic index names using * as a wildcard. Example: logstash-*

☐ Do not expand index pattern when searching (Not recommended)

By default, searches against any time-based index pattern that contains a wildcard will automatically be expanded to query only the indices that contain data within the currently selected time range.

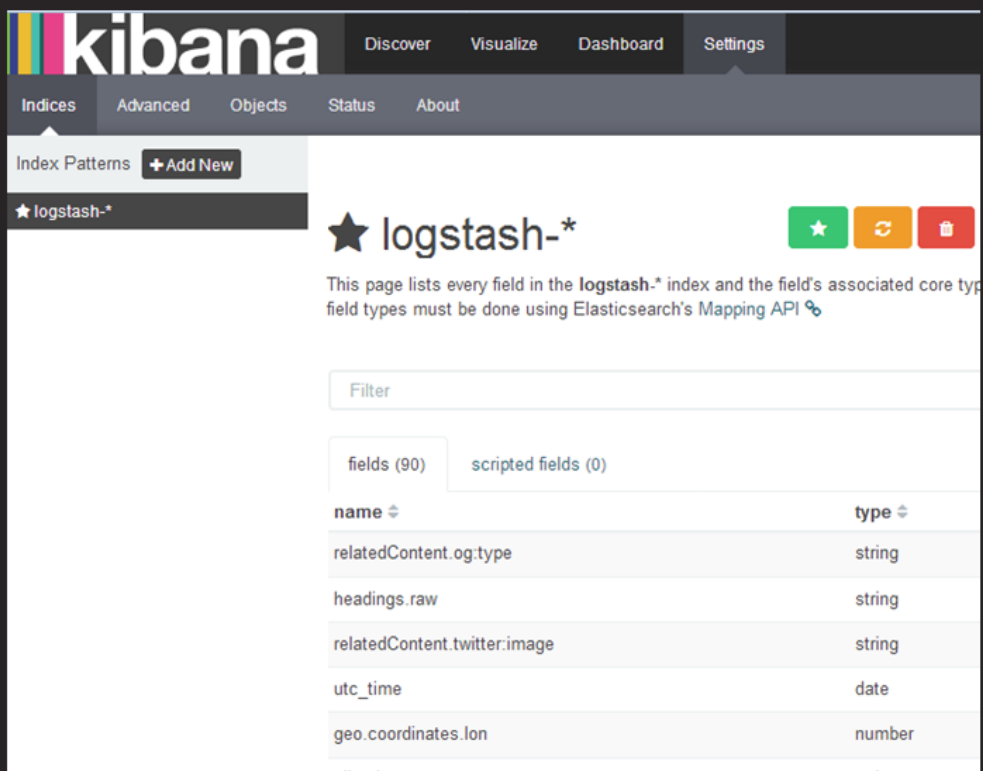
Searching against the index pattern `logstash-*` will actually query elasticsearch for the specific matching indices (e.g. `logstash-2015.12.21`) that fall within the current time range.

Time-field name ⓘ refresh fields



Kibana 部分 (续4)

知识讲解



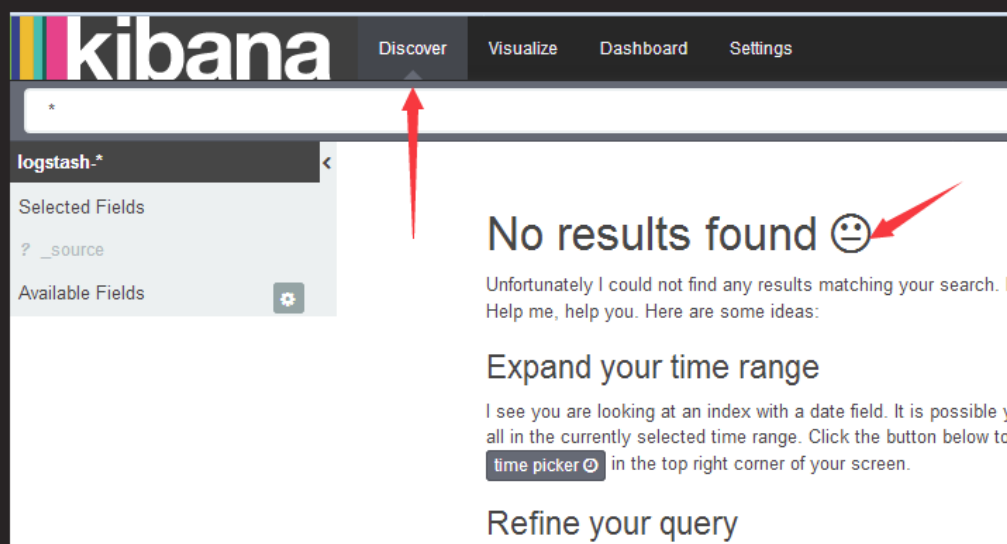
The screenshot shows the Kibana interface with the 'Indices' tab selected. Under 'Index Patterns', the 'logstash-*' pattern is highlighted. The main content area displays the fields for this index pattern. A table lists the fields and their types:

name	type
relatedContent.og:type	string
headings.raw	string
relatedContent.twitter:image	string
utc_time	date
geo.coordinates.lon	number



Kibana 部分 (续5)

- 导入成功以后选择Discover



知识讲解



Kibana 部分 (续6)

- 这里没有数据的原因是我们导入的日志是2015-05-10至2015-05-20的时间段，默认配置是最近15分钟，在这可以修改一下时间来显示

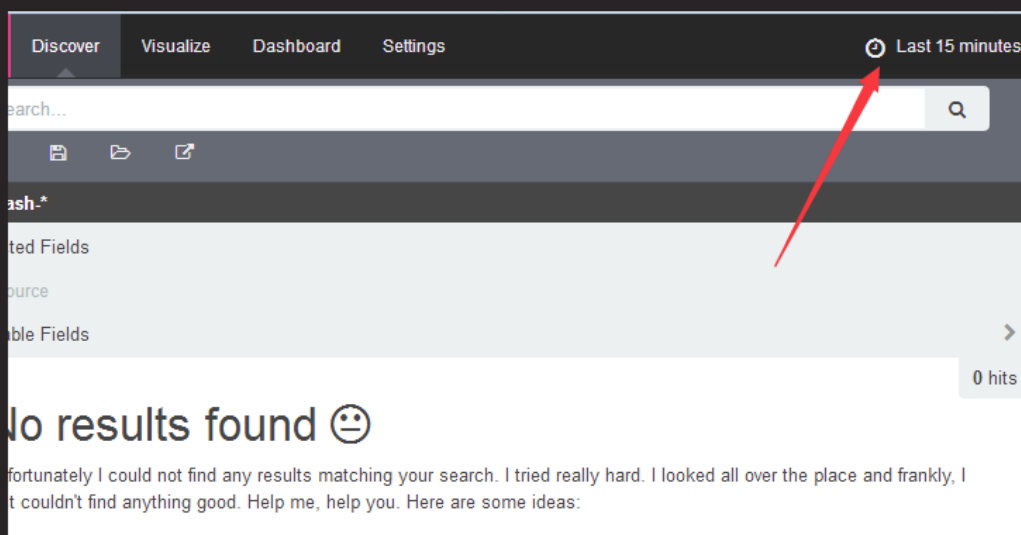
知识讲解



Kibana修改时间

- 修改时间

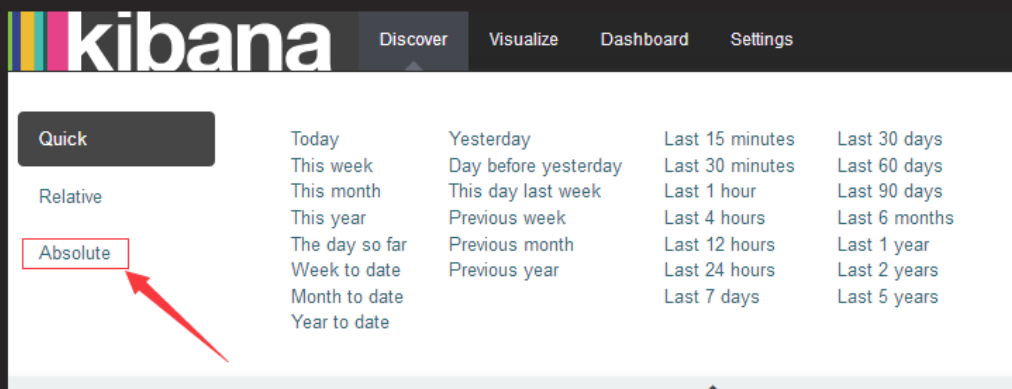
知识讲解



Kibana修改时间（续1）

- 修改时间

知识讲解



Kibana修改时间（续2）

- 修改时间

知识讲解

From: 2015-05-12 20:21:43.189
YYYY-MM-DD HH:mm:ss.SSS

To: Set To Now 2015-05-22 20:36:43.191
YYYY-MM-DD HH:mm:ss.SSS

Go

May 2015

Sun	Mon	Tue	Wed	Thu	Fri	Sat
26	27	28	29	30	01	02
03	04	05	06	07	08	09
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	01	02	03	04	05	06

May 2015

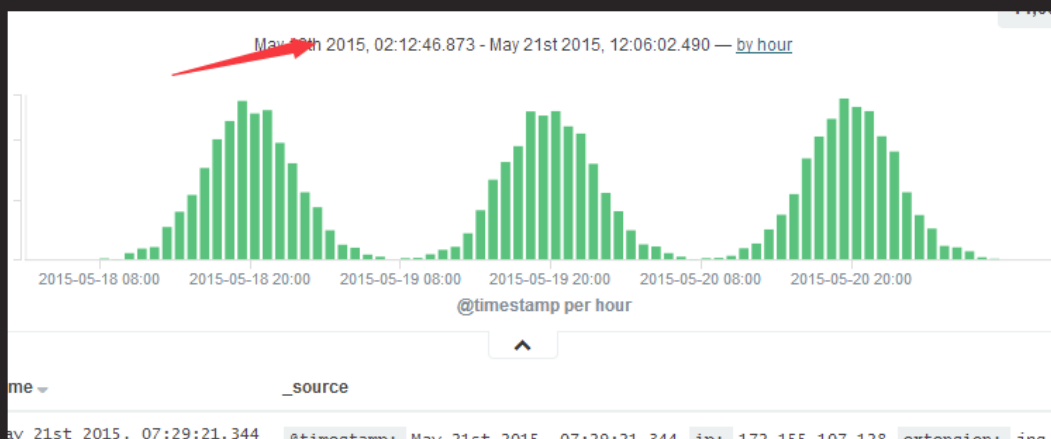
Sun	Mon	Tue	Wed	Thu	Fri	Sat
26	27	28	29	30	01	02
03	04	05	06	07	08	09
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	01	02	03	04	05	06



Kibana数据展示

- 数据展示

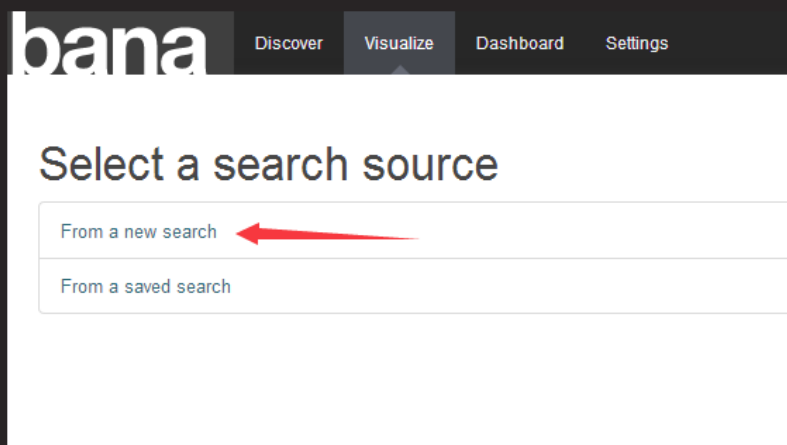
知识讲解



Kibana展示方式（续1）

- 饼图与列表，多种维度自定义统计分析

知识讲解



Kibana展示方式（续2）

- 饼图与列表，多种维度自定义统计分析

知识讲解

The screenshot shows the Kibana configuration interface for a visualization. The 'metrics' section at the top has 'Slice Size' selected with a 'Count' metric. Below it, the 'buckets' section has 'Split Slices' selected in the 'Select buckets type' dropdown. A red arrow points to the 'Split Slices' option. A 'Cancel' button is visible at the bottom of the buckets panel.



Kibana展示方式（续3）

- 饼图与列表，多种维度自定义统计分析

知识讲解

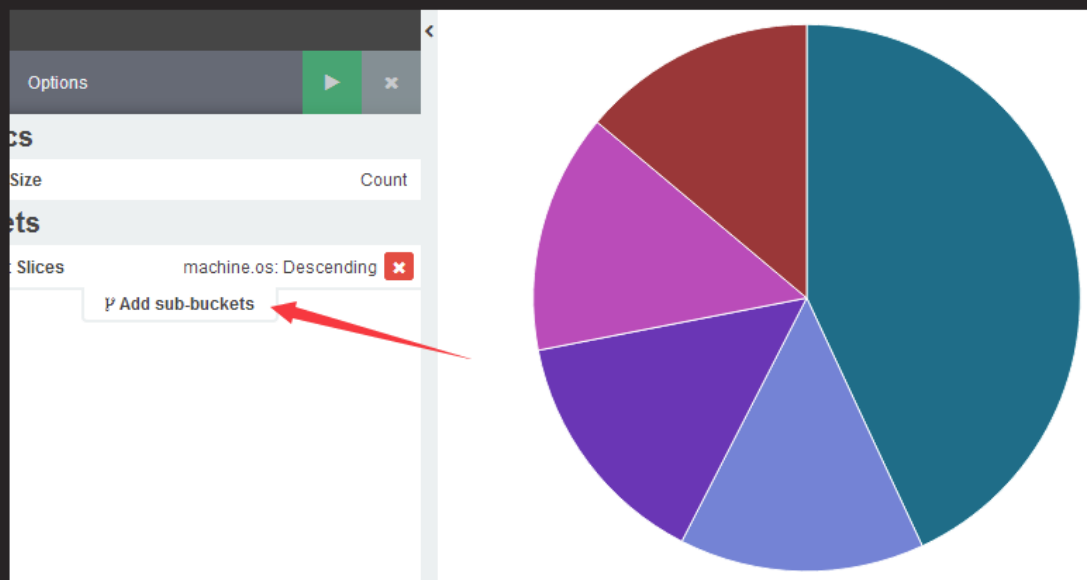
The screenshot shows the 'buckets' configuration panel with 'Split Slices' expanded. A red box highlights the 'Field' input field. Red arrows point to the 'Terms' aggregation dropdown, the 'Field' input field, and the 'Order By' dropdown menu. The 'Order By' menu is currently set to 'metric: Count'. The 'Order' dropdown is set to 'Descending' and the 'Size' is set to '5'. A 'CustomLabel' field is visible at the bottom.



Kibana展示方式（续4）

- 饼图与列表，多种维度自定义统计分析

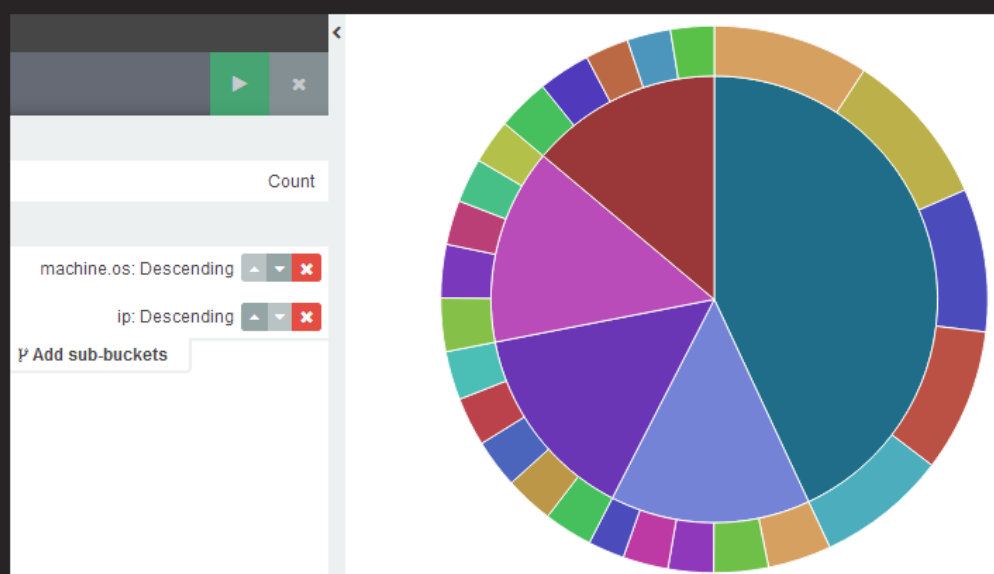
知识讲解



Kibana展示方式（续5）

- 饼图与列表，多种维度自定义统计分析

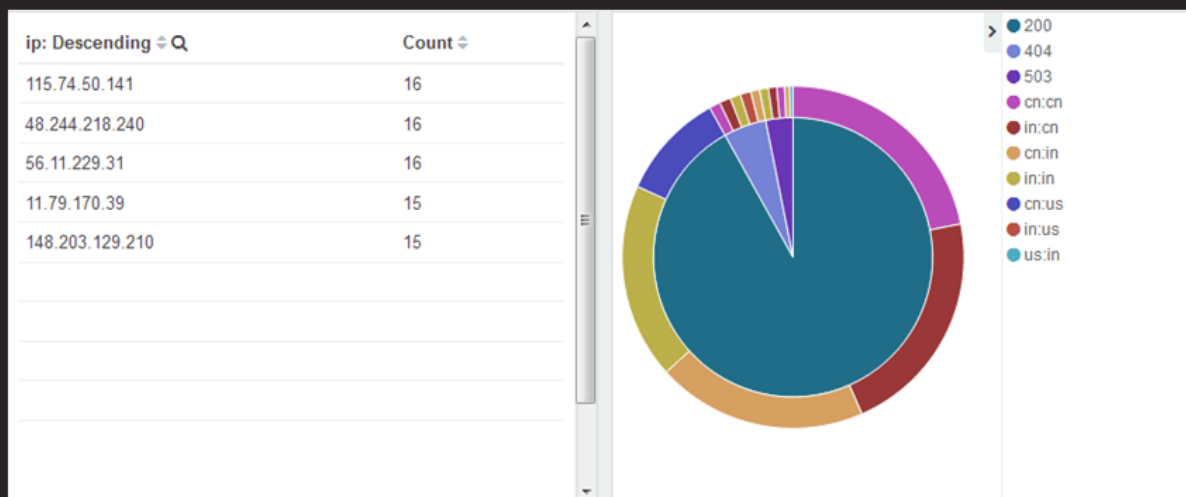
知识讲解



Kibana展示方式（续6）

- 列表与饼图

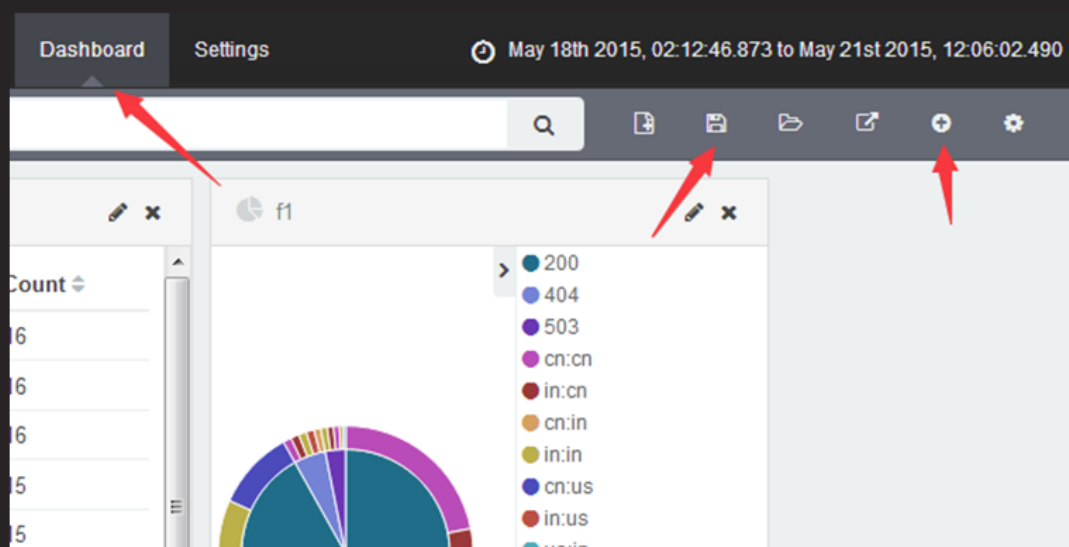
知识讲解



Kibana展示方式（续7）

- 保存后可以在Dashboard查看

知识讲解



Logstash配置扩展插件

Logstash配置扩展插件

Logstash

Logstash是什么

Logstash安装

Logstash类型及条件判断

Logstash配置文件

插件

filebeat安装配置

Logstash是什么

知识讲解

- 是一个数据采集、加工处理以及传输的工具
- 特点
 - 所有类型的数据集中处理
 - 不同模式和格式数据的正常化
 - 自定义日志格式的迅速扩展
 - 为自定义数据源轻松添加插件



Logstash安装 (续1)

- Logstash工作结构

知识讲解

```
{ 数据源 } ==>
      input { } ==>
            filter { } ==>
                  output { } ==>
                        { ES }
```



Logstash类型及条件判断

- Logstash里面的类型
 - 布尔值类型: `ssl_enable => true`
 - 字节类型: `bytes => "1MiB"`
 - 字符串类型: `name => "xkops"`
 - 数值类型: `port => 22`
 - 数组: `match => ["datetime","UNIX"]`
 - 哈希: `options => {k => "v",k2 => "v2"}`
 - 编码解码: `codec => "json"`
 - 路径: `file_path => "/tmp/filename"`
 - 注释: `#`

知识讲解



Logstash类型及条件判断（续1）

知识讲解

- Logstash条件判断

- 等于: ==
- 不等于: !=
- 小于: <
- 大于: >
- 小于等于: <=
- 大于等于: >=
- 匹配正则: =~
- 不匹配正则: !~



Logstash类型及条件判断（续2）

知识讲解

- Logstash条件判断

- 包含: in
- 不包含: not in
- 与: and
- 或: or
- 非与: nand
- 非或: xor
- 复合表达式: ()
- 取反符合: !()



Logstash配置文件

知识讲解

- Logstash的第一个配置文件
 - /etc/logstash/logstash.conf

```
input{
  stdin{}
}
filter{ }
output{
  stdout{}
}
```

- 启动并验证

```
# logstash -f logstash.conf
```



插件

知识讲解

- Logstash插件
 - 上页的配置文件使用了logstash-input-stdin和logstash-output-stdout两个插件，Logstash还有filter和codec类插件，查看插件的方式是

```
# /opt/logstash/bin/logstash-plugin list
```
 - 插件及文档地址
<https://github.com/logstash-plugins>
- 练习
 - Logstash配置从标准输入读取输入源，然后将结果输出到屏幕



插件（续1）

知识讲解

- codec类插件
 - 常用的插件：plain、json、json_lines、rubydebug、multiline等
 - 使用刚刚的例子，这次输入json数据
 - 设置输入源的codec是json，在输入的时候选择rubydebug



插件（续2）

知识讲解

- codec类插件

```
input{
  stdin{ codec => "json" }
}
filter{ }
output{
  stdout{ codec => "rubydebug" }
}
```
- 输入普通数据和json对比

```
{"a": 1, "c": 3, "b": 2}
```



插件（续3）

知识讲解

- codec类插件
 - 练习output和input配置
 - 练习在input不指定类型json输出结果
 - 练习在output不指定rubydebug的输出结果
 - 同时指定以后的输出结果



插件（续4）

知识讲解

- 练习input file插件

```
file{
    start_position => "beginning"
    sincedb_path => "/var/lib/logstash/sincedb-access"
    path => [ "/tmp/alog" , "/tmp/blog" ]
    type => 'filelog'
}
```

 - sincedb_path记录读取文件的位置
 - start_position配置第一次读取文件从什么地方开始



插件（续5）

- 练习input tcp和udp插件

```
tcp{
    host => "0.0.0.0"
    port => 8888
    type => "tcplog"
}
udp{
    host => "192.168.4.16"
    port => 9999
    type => "udplog"
}
```

知识讲解



