

课程介绍

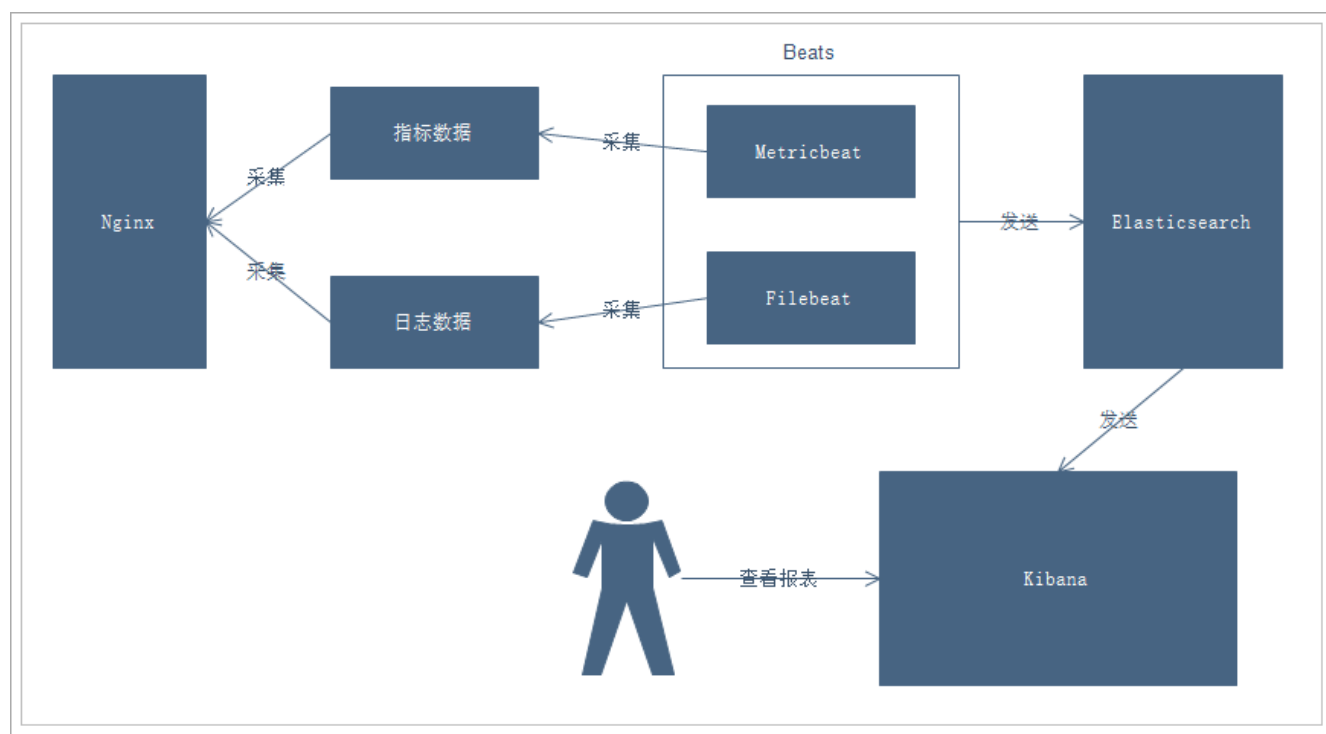
- Nginx日志分析系统
- Filebeat入门学习
- Metricbeat入门学习
- Kibana入门学习
- Logstash入门学习
- 综合练习

1、Nginx日志分析系统

1.1、项目需求

Nginx是一款非常优秀的web服务器，往往nginx服务会作为项目的访问入口，那么，nginx的性能保障就变得非常重要了，如果nginx的运行出现了问题就会对项目有较大的影响，所以，我们需要对nginx的运行有监控措施，实时掌握nginx的运行情况，那就需要收集nginx的运行指标和分析nginx的运行日志了。

1.2、业务流程



说明：

- 通过Beats采集Nginx的指标数据和日志数据
- Beats采集到数据后发送到Elasticsearch中
- Kibana读取数据进行分析
- 用户通过Kibana进行查看分析报表

2、部署安装Nginx

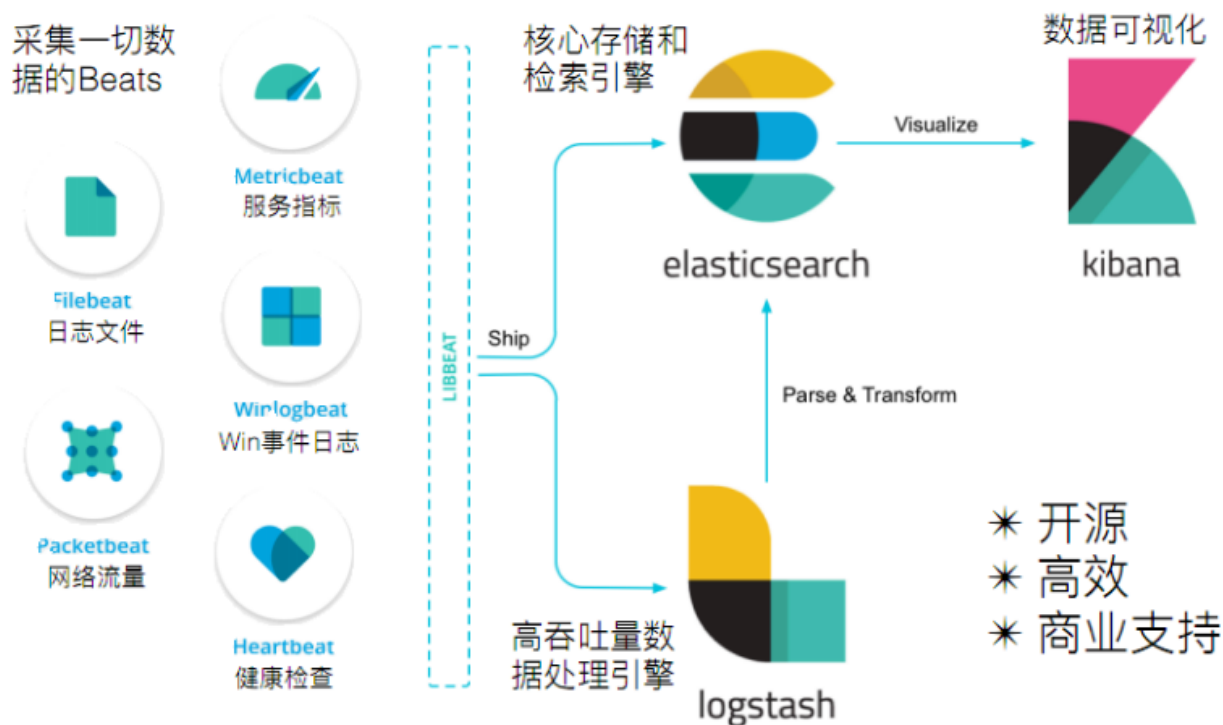


```
1 tar -xvf nginx-1.11.6.tar.gz
2 yum -y install pcre-devel zlib-devel
3 ./configure
4 make install
5 #启动
6 cd /usr/local/nginx/sbin/
7 ./nginx
8
9 #通过浏览器访问页面并且查看日志
10 #访问地址: http://192.168.40.133/
11 tail -f /usr/local/nginx/logs/access.log
```

```
[root@node01 logs]# tail /usr/local/nginx/logs/access.log
192.168.40.1 - - [18/Feb/2019:10:38:22 +0800] "GET / HTTP/1.1" 200 612 "-" "Mozilla/5.0 (Windows NT 6.3; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.67 Safari/537.36"
192.168.40.1 - - [18/Feb/2019:10:38:22 +0800] "GET /favicon.ico HTTP/1.1" 404 571 "http://192.168.40.133/" "Mozilla/5.0 (Windows NT 6.3; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.67 Safari/537.36"
192.168.40.1 - - [18/Feb/2019:10:38:24 +0800] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Windows NT 6.3; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.67 Safari/537.36"
192.168.40.1 - - [18/Feb/2019:10:41:07 +0800] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Windows NT 6.3; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.67 Safari/537.36"
```

3、Beats 简介

ElasticStack的组成



官网：<https://www.elastic.co/cn/products/beats>



轻量型数据采集器

Beats 平台集合了多种单一用途数据采集器。它们从成百上千或成千上万台机器和系统向 Logstash 或 Elasticsearch 发送数据。

Beats系列产品：

Beats 系列

全品类采集器，搞定所有数据类型。



Filebeat
日志文件



Metricbeat
指标



Packetbeat
网络数据



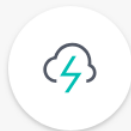
Winlogbeat
Windows 事件日志



Auditbeat
审计数据

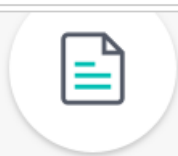


Heartbeat
运行时间监控



Functionbeat
无需服务器的采集器

4、Filebeat



Filebeat

轻量型日志采集器

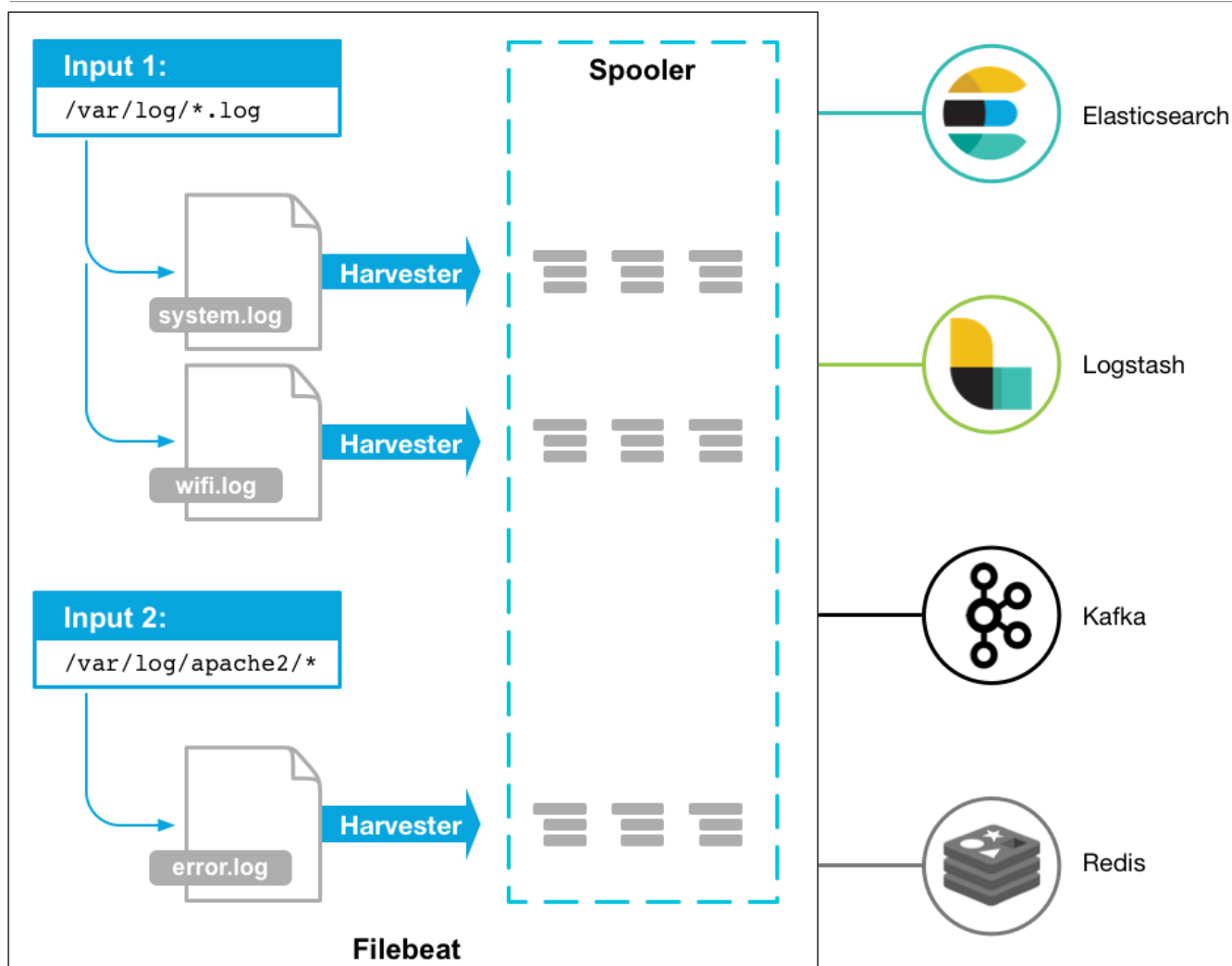
当您要面对成百上千、甚至成千上万的服务器、虚拟机和容器生成的日志时，请告别 SSH 吧。Filebeat 将为您提供一种轻量型方法，用于转发和汇总日志与文件，让简单的事情不再繁杂。

汇总、“tail -f” 和搜索

启动 Filebeat 后，打开 Logs UI，直接在 Kibana 中观看对您的文件进行 tail 操作的过程。通过搜索栏按照服务、应用程序、主机、数据中心或者其他条件进行筛选，以跟踪您的全部汇总日志中的异常行为。

4.1、架构

用于监控、收集服务器日志文件。



4.2、部署与运行

下载（或使用资料中提供的安装包，版本为：filebeat-6.5.4）：<https://www.elastic.co/downloads/beats>

```
1 mkdir /itcast/beats
2 tar -xvf filebeat-6.5.4-linux-x86_64.tar.gz
3 cd filebeat-6.5.4-linux-x86_64
4
5 #创建如下配置文件 itcast.yml
6 filebeat.inputs:
7 - type: stdin
8   enabled: true
9 setup.template.settings:
10   index.number_of_shards: 3
11 output.console:
12   pretty: true
13   enable: true
14
15 #启动filebeat
16 ./filebeat -e -c itcast.yml
17
18 #输入hello运行结果如下：
19 hello
```



```
20 {
21   "@timestamp": "2019-01-12T12:50:03.585z",
22   "@metadata": { #元数据信息
23     "beat": "filebeat",
24     "type": "doc",
25     "version": "6.5.4"
26   },
27   "source": "",
28   "offset": 0,
29   "message": "hello", #输入的内容
30   "prospector": { #标准输入勘探器
31     "type": "stdin"
32   },
33   "input": { #控制台标准输入
34     "type": "stdin"
35   },
36   "beat": { #beat版本以及主机信息
37     "name": "itcast01",
38     "hostname": "itcast01",
39     "version": "6.5.4"
40   },
41   "host": {
42     "name": "itcast01"
43   }
44 }
45
```

4.3、读取文件

```
1 #配置读取文件项 itcast-log.yml
2
3 filebeat.inputs:
4 - type: log
5   enabled: true
6   paths:
7     - /itcast/beats/logs/*.log
8 setup.template.settings:
9   index.number_of_shards: 3
10 output.console:
11   pretty: true
12   enable: true
13
14 #启动filebeat
15 ./filebeat -e -c itcast-log.yml
16
17 #/haoke/beats/logs下创建a.log文件，并输入如下内容
18 hello
19 world
20
21 #观察filebeat输出
22 {
23   "@timestamp": "2019-01-12T14:16:10.192z",
```



```
24  "@metadata": {
25      "beat": "filebeat",
26      "type": "doc",
27      "version": "6.5.4"
28  },
29  "host": {
30      "name": "itcast01"
31  },
32  "source": "/haoke/beats/logs/a.log",
33  "offset": 0,
34  "message": "hello",
35  "prospector": {
36      "type": "log"
37  },
38  "input": {
39      "type": "log"
40  },
41  "beat": {
42      "version": "6.5.4",
43      "name": "itcast01",
44      "hostname": "itcast01"
45  }
46  }
47  {
48      "@timestamp": "2019-01-12T14:16:10.192Z",
49      "@metadata": {
50          "beat": "filebeat",
51          "type": "doc",
52          "version": "6.5.4"
53      },
54      "prospector": {
55          "type": "log"
56      },
57      "input": {
58          "type": "log"
59      },
60      "beat": {
61          "version": "6.5.4",
62          "name": "itcast01",
63          "hostname": "itcast01"
64      },
65      "host": {
66          "name": "itcast01"
67      },
68      "source": "/haoke/beats/logs/a.log",
69      "offset": 6,
70      "message": "world"
71  }
72
```

可以看出，已经检测到日志文件有更新，立刻就会读取到更新的内容，并且输出到控制台。

4.4、自定义字段

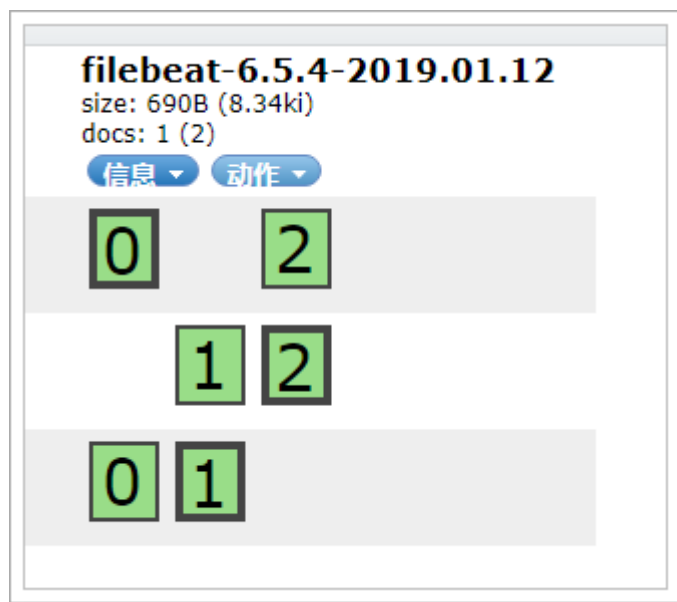


```
1 #配置读取文件项 itcast-log.yml
2
3 filebeat.inputs:
4 - type: log
5   enabled: true
6   paths:
7     - /itcast/beats/logs/*.log
8   tags: ["web"] #添加自定义tag, 便于后续的处理
9   fields: #添加自定义字段
10     from: itcast-im
11     fields_under_root: true #true为添加到根节点, false为添加到子节点中
12 setup.template.settings:
13   index.number_of_shards: 3
14 output.console:
15   pretty: true
16   enable: true
17
18 #启动filebeat
19 ./filebeat -e -c itcast-log.yml
20
21 #/haoke/beats/logs下创建a.log文件, 并输入如下内容
22 123
23
24 #执行效果
25 {
26   "@timestamp": "2019-01-12T14:37:19.845Z",
27   "@metadata": {
28     "beat": "filebeat",
29     "type": "doc",
30     "version": "6.5.4"
31   },
32   "offset": 0,
33   "tags": [
34     "haoke-im"
35   ],
36   "prospector": {
37     "type": "log"
38   },
39   "beat": {
40     "name": "itcast01",
41     "hostname": "itcast01",
42     "version": "6.5.4"
43   },
44   "host": {
45     "name": "itcast01"
46   },
47   "source": "/itcast/beats/logs/a.log",
48   "message": "123",
49   "input": {
50     "type": "log"
51   },
52   "from": "haoke-im"
53 }
```

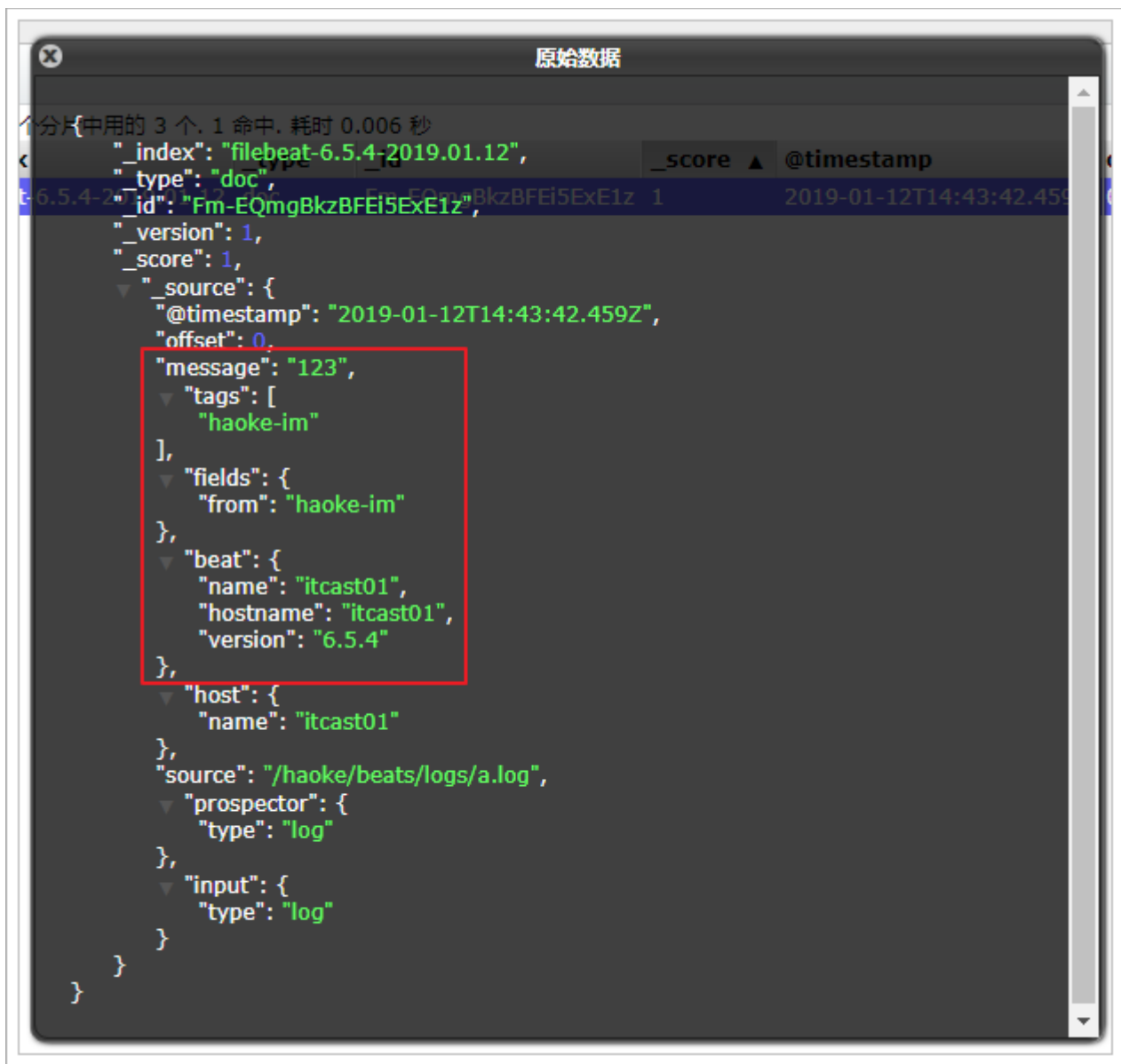

4.5、输出到Elasticsearch

```
1 # itcast-log.yml
2 filebeat.inputs:
3   - type: log
4     enabled: true
5     paths:
6       - /itcast/beats/logs/*.log
7     tags: ["haoke-im"]
8     fields:
9       from: haoke-im
10    fields_under_root: false
11  setup.template.settings:
12    index.number_of_shards: 3 #指定索引的分区数
13  output.elasticsearch: #指定ES的配置
14    hosts: ["192.168.1.7:9200", "192.168.1.7:9201", "192.168.1.7:9202"]
```

在日志文件中输入新的内容进行测试：



查看数据：



4.6、Filebeat工作原理

Filebeat由两个主要组件组成：prospector 和 harvester。

- harvester :
 - 负责读取单个文件的内容。
 - 如果文件在读取时被删除或重命名，Filebeat将继续读取文件。
- prospector
 - prospector 负责管理harvester并找到所有要读取的文件来源。
 - 如果输入类型为日志，则查找器将查找路径匹配的所有文件，并为每个文件启动一个harvester。
 - Filebeat目前支持两种prospector类型：log和stdin。
- Filebeat如何保持文件的状态
 - Filebeat 保存每个文件的状态并经常将状态刷新到磁盘上的注册文件中。
 - 该状态用于记住harvester正在读取的最后偏移量，并确保发送所有日志行。
 - 如果输出（例如Elasticsearch或Logstash）无法访问，Filebeat会跟踪最后发送的行，并在输出再次可用时继续读取文件。

- 在Filebeat运行时，每个prospector内存中也会保存的文件状态信息，当重新启动Filebeat时，将使用注册文件的数据来重建文件状态，Filebeat将每个harvester在从保存的最后偏移量继续读取。
- 文件状态记录在data/registry文件中。

启动命令：

```
1 ./filebeat -e -c itcast.yml
2 ./filebeat -e -c itcast.yml -d "publish"
3
4 #参数说明
5 -e: 输出到标准输出，默认输出到syslog和logs下
6 -c: 指定配置文件
7 -d: 输出debug信息
8
9 #测试: ./filebeat -e -c itcast-log.yml -d "publish"
10 DEBUG [publish] pipeline/processor.go:308 Publish event: {
11   "@timestamp": "2019-01-12T15:03:50.820Z",
12   "@metadata": {
13     "beat": "filebeat",
14     "type": "doc",
15     "version": "6.5.4"
16   },
17   "offset": 0,
18   "tags": [
19     "haoke-im"
20   ],
21   "input": {
22     "type": "log"
23   },
24   "prospector": {
25     "type": "log"
26   },
27   "beat": {
28     "name": "itcast01",
29     "hostname": "itcast01",
30     "version": "6.5.4"
31   },
32   "source": "/haoke/beats/logs/a.log",
33   "fields": {
34     "from": "haoke-im"
35   },
36   "host": {
37     "name": "itcast01"
38   },
39   "message": "456"
40 }
41
```

4.7、读取Nginx日志文件



```

1 # itcast-nginx.yml
2 filebeat.inputs:
3   - type: log
4     enabled: true
5     paths:
6       - /usr/local/nginx/logs/*.log
7     tags: ["nginx"]
8 setup.template.settings:
9   index.number_of_shards: 3 #指定索引的分区数
10 output.elasticsearch: #指定ES的配置
11   hosts: ["192.168.40.133:9200", "192.168.40.134:9200", "192.168.40.135:9200"]

```

```

1 #启动
2 ./filebeat -e -c itcast-nginx.yml

```



启动后，可以在Elasticsearch中看到索引以及查看数据：

查询 8 个分片中用的 8 个, 9 命中, 耗时 0.020 秒

_index	_type	_id	_score ▲	@timestamp	beat.version	beat.name	beat.hostname	host.n
filebeat-6.5.4-2019.03.14	doc	k2D-e2kBkOyweKwjBCKP	1	2019-03-14T11:37:15.036Z	6.5.4	node01	node01	node01
filebeat-6.5.4-2019.03.14	doc	i2D-e2kBkOyweKwjBCKP	1	2019-03-14T11:37:15.036Z	6.5.4	node01	node01	node01
filebeat-6.5.4-2019.03.14	doc	jGD-e2kBkOyweKwjBCKP	1	2019-03-14T11:37:15.036Z	6.5.4	node01	node01	node01
filebeat-6.5.4-2019.03.14	doc	j2D-e2kBkOyweKwjBCKP	1	2019-03-14T11:37:15.036Z	6.5.4	node01	node01	node01
filebeat-6.5.4-2019.03.14	doc	kGD-e2kBkOyweKwjBCKP	1	2019-03-14T11:37:15.036Z	6.5.4	node01	node01	node01
filebeat-6.5.4-2019.03.14	doc	kWD-e2kBkOyweKwjBCKP	1	2019-03-14T11:37:15.036Z	6.5.4	node01	node01	node01
filebeat-6.5.4-2019.03.14	doc	kmD-e2kBkOyweKwjBCKP	1	2019-03-14T11:37:15.036Z	6.5.4	node01	node01	node01
filebeat-6.5.4-2019.03.14	doc	jWD-e2kBkOyweKwjBCKP	1	2019-03-14T11:37:15.036Z	6.5.4	node01	node01	node01
filebeat-6.5.4-2019.03.14	doc	jmD-e2kBkOyweKwjBCKP	1	2019-03-14T11:37:15.036Z	6.5.4	node01	node01	node01



```
原始数据
{"_type": "doc", "index": "filebeat-6.5.4-2019.03.14", "id": "k2D-e2kBkOyweKwjBCKP", "score": 1, "source": {"@timestamp": "2019-03-14T11:37:15.036Z", "beat": {"name": "node01", "hostname": "node01"}, "host": {"name": "node01"}, "source": "/usr/local/nginx/logs/access.log", "offset": 21652, "message": "192.168.40.1 - - [14/Mar/2019:19:32:01 +0800] \"GET / HTTP/1.1\" 304 0 \"-\" \"Mozilla/5.0 (Windows NT 6.3; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.67 Safari/537.36\"", "tags": ["nginx"], "prospector": {"type": "log"}, "input": {"type": "log"}}, "type": "log", "type": "log"}}
```

可以看到，在message中已经获取到了nginx的日志，但是，内容并没有经过处理，只是读取到原数据，那么对于我们后期的操作是不利的，有办法解决吗？

4.7、Module

前面要想实现日志数据的读取以及处理都是自己手动配置的，其实，在Filebeat中，有大量的Module，可以简化我们的配置，直接就可以使用，如下：

```
1 ./filebeat modules list
2
3 Enabled:
4
5 Disabled:
6 apache2
7 auditd
8 elasticsearch
9 haproxy
10 icinga
11 iis
12 kafka
```

```
13 kibana
14 logstash
15 mongodb
16 mysql
17 nginx
18 osquery
19 postgresql
20 redis
21 suricata
22 system
23 traefik
24
```

可以看到，内置了很多的module，但是都没有启用，如果需要启用需要进行enable操作：

```
1 ./filebeat modules enable nginx #启动
2 ./filebeat modules disable nginx #禁用
3
4 Enabled:
5 nginx
6
7 Disabled:
8 apache2
9 auditd
10 elasticsearch
11 haproxy
12 icinga
13 iis
14 kafka
15 kibana
16 logstash
17 mongodb
18 mysql
19 redis
20 osquery
21 postgresql
22 suricata
23 system
24 traefik
```

可以发现，nginx的module已经被启用。

4.7.1、nginx module 配置

```
1 - module: nginx
2   # Access logs
3   access:
4     enabled: true
5     var.paths: ["/usr/local/nginx/logs/access.log*"]
6
7   # Set custom paths for the log files. If left empty,
8   # Filebeat will choose the paths depending on your OS.
```



```
9      #var.paths:
10
11      # Error logs
12      error:
13          enabled: true
14          var.paths: ["/usr/local/nginx/logs/error.log*"]
15
16          # Set custom paths for the log files. If left empty,
17          # Filebeat will choose the paths depending on your OS.
18      #var.paths:
```

4.7.2、配置filebeat

```
1  #vim itcast-nginx.yml
2
3  filebeat.inputs:
4      #- type: log
5      #   enabled: true
6      #   paths:
7      #       - /usr/local/nginx/logs/*.log
8      #   tags: ["nginx"]
9  setup.template.settings:
10     index.number_of_shards: 3
11  output.elasticsearch:
12     hosts: ["192.168.40.133:9200", "192.168.40.134:9200", "192.168.40.135:9200"]
13  filebeat.config.modules:
14     path: ${path.config}/modules.d/*.yaml
15     reload.enabled: false
```

4.7.3、测试

```
1  ./filebeat -e -c itcast-nginx.yml
2
3  #启动会出错，如下
4  ERROR   fileset/factory.go:142  Error loading pipeline: Error loading pipeline for
      fileset nginx/access: This module requires the following Elasticsearch plugins:
      ingest-user-agent, ingest-geoip. You can install them by running the following
      commands on all the Elasticsearch nodes:
5      sudo bin/elasticsearch-plugin install ingest-user-agent
6      sudo bin/elasticsearch-plugin install ingest-geoip
7
8  #解决：需要在Elasticsearch中安装ingest-user-agent、ingest-geoip插件
9  #在资料中可以找到，ingest-user-agent.tar、ingest-geoip.tar、ingest-geoip-conf.tar 3个文件
10 #其中，ingest-user-agent.tar、ingest-geoip.tar解压到plugins下
11 #ingest-geoip-conf.tar解压到config下
12 #问题解决。
```



查询 3 个分片中用的 3 个, 20 命中, 耗时 0.147 秒

_index	_type	_id	_score	offset	nginx.access.referrer	nginx.access.response_code	nginx.access.re
filebeat-6.5.4-2019.03.15	doc	QpN1fGkBNS8mqcrlWpaQ	1	25054	-	304	192.168.40.1
filebeat-6.5.4-2019.03.15	doc	Rfn1fGkBNS8mqcrlWpaQ	1	25999	-	304	192.168.40.1
filebeat-6.5.4-2019.03.15	doc	w1b5fGkBThZl6kQqp4ft	1	27133	-	304	192.168.40.1
filebeat-6.5.4-2019.03.15	doc	wob5fGkBThZl6kQqp4ft	1	27511	-	304	192.168.40.1
filebeat-6.5.4-2019.03.15	doc	w4b5fGkBThZl6kQqp4ft	1	27700	-	304	192.168.40.1
filebeat-6.5.4-2019.03.15	doc	Pvn1fGkBNS8mqcrlWpaQ	1	24676	-	304	192.168.40.1
filebeat-6.5.4-2019.03.15	doc	Qvn1fGkBNS8mqcrlWpaQ	1	25432	-	304	192.168.40.1
filebeat-6.5.4-2019.03.15	doc	Q_n1fGkBNS8mqcrlWpaQ	1	25621	-	304	192.168.40.1
filebeat-6.5.4-2019.03.15	doc	efH3fGkBv1kB6V7tSoQu	1	1019	-	-	-
filebeat-6.5.4-2019.03.15	doc	vob3fGkBThZl6kQqhYch	1	26566	-	304	192.168.40.1
filebeat-6.5.4-2019.03.15	doc	v4b3fGkBThZl6kQqhYch	1	26755	-	304	192.168.40.1
filebeat-6.5.4-2019.03.15	doc	evH5fGkBv1kB6V7tboQp	1	26944	-	304	192.168.40.1
filebeat-6.5.4-2019.03.15	doc	wYb5fGkBThZl6kQqp4ft	1	27322	-	304	192.168.40.1
filebeat-6.5.4-2019.03.15	doc	P_n1fGkBNS8mqcrlWpaQ	1	24865	-	304	192.168.40.1
filebeat-6.5.4-2019.03.15	doc	Qfn1fGkBNS8mqcrlWpaQ	1	25243	-	304	192.168.40.1
filebeat-6.5.4-2019.03.15	doc	Rpn1fGkBNS8mqcrlWpaQ	1	25810	-	304	192.168.40.1
filebeat-6.5.4-2019.03.15	doc	v1b3fGkBThZl6kQqhYch	1	26188	-	304	192.168.40.1
filebeat-6.5.4-2019.03.15	doc	vYb3fGkBThZl6kQqhYch	1	26377	-	304	192.168.40.1
filebeat-6.5.4-2019.03.15	doc	m6D5fGkBGPQwA_tdXOs6	1	1079	-	-	-
filebeat-6.5.4-2019.03.15	doc	x1b5fGkBThZl6kQqp4ft	1	27889	-	304	192.168.40.1

测试发现，数据已经写入到了Elasticsearch中，并且拿到的数据更加明确了：

```

2019.03.15 doc "bytes": "0" 5fGkBThZl6kQqp4ft 1 27700 -
2019.03.15 doc Pvn1fGkBNS8mqcrlWpaQ 1 24676 -
2019.03.15 doc "remote_ip_list": [
2019.03.15 doc "192.168.40.1" kBNS8mqcrlWpaQ 1 25432 -
2019.03.15 doc Q_n1fGkBNS8mqcrlWpaQ 1 25621 -
2019.03.15 doc efH3fGkBv1kB6V7tSoQu 1 1019
2019.03.15 doc vob3fGkBThZl6kQqhYch 1 26566 -
2019.03.15 doc "patch": "3538", 24865 -
2019.03.15 doc "original": "Mozilla/5.0 (Windows NT 6.3; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.67
Safari/537.36", 26944 -
2019.03.15 doc "major": "70", GkBThZl6kQqp4ft 1 27322 -
2019.03.15 doc "minor": "0", P_n1fGkBNS8mqcrlWpaQ 1 24865 -
2019.03.15 doc "os": "Windows 8.1", Qfn1fGkBNS8mqcrlWpaQ 1 25243 -
2019.03.15 doc "name": "Chrome", Rpn1fGkBNS8mqcrlWpaQ 1 25810 -
2019.03.15 doc "os_name": "Windows 8.1", v1b3fGkBThZl6kQqhYch 1 26188 -
2019.03.15 doc "device": "Other", vYb3fGkBThZl6kQqhYch 1 26377 -

```

当然了，其他的Module的用法参加官方文档：

<https://www.elastic.co/guide/en/beats/filebeat/current/filebeat-modules.html>

- [Modules overview](#)
- [Apache2 module](#)
- [Auditd module](#)
- [Elasticsearch module](#)
- [haproxy module](#)
- [Icinga module](#)
- [IIS module](#)
- [Kafka module](#)
- [Kibana module](#)
- [Logstash module](#)
- [MongoDB module](#)
- [MySQL module](#)
- [Nginx module](#)
- [Osquery module](#)
- [PostgreSQL module](#)
- [Redis module](#)
- [Suricata module](#)
- [System module](#)

5、Metricbeat



Metricbeat

轻量型指标采集器

用于从系统和服务收集指标。Metricbeat 能够以一种轻量型的方式，输送各种系统和服务统计数据，从 CPU 到内存，从 Redis 到 Nginx，不一而足。

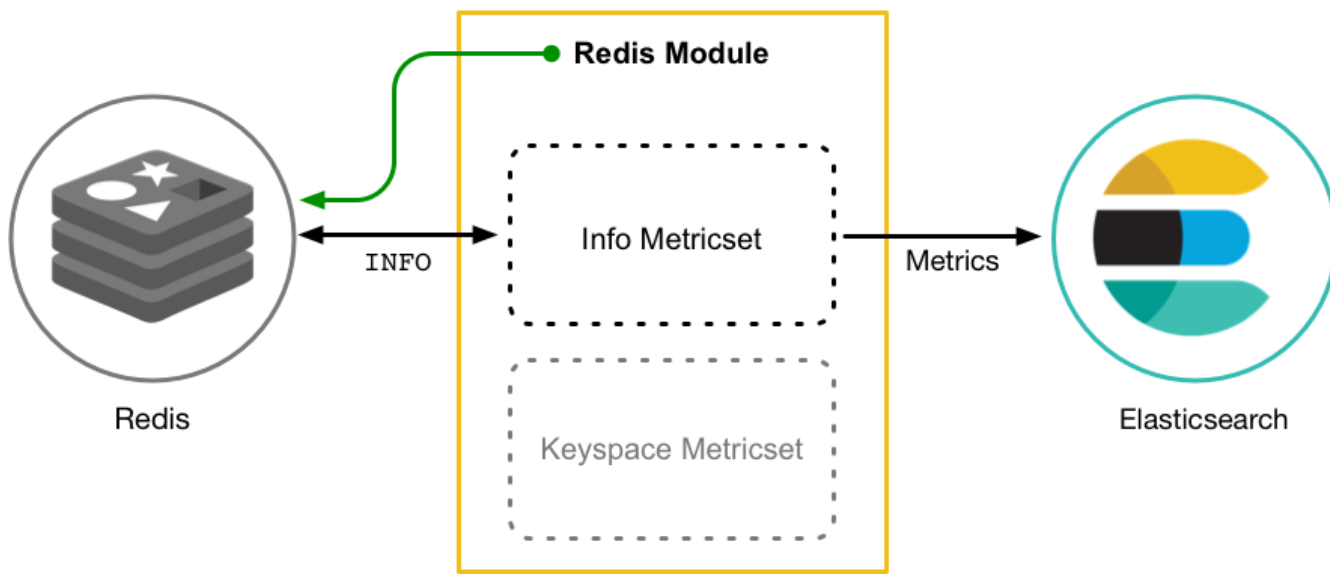
- 定期收集操作系统或应用服务的指标数据
- 存储到Elasticsearch中，进行实时分析

5.1、Metricbeat组成

Metricbeat有2部分组成，一部分是Module，另一部分为Metricset。

- Module
 - 收集的对象，如：mysql、redis、nginx、操作系统等；
- Metricset
 - 收集指标的集合，如：cpu、memory、network等；

以Redis Module为例：



5.2、部署与收集系统指标

```
1 tar -xvf metricbeat-6.5.4-linux-x86_64.tar.gz
2 cd metricbeat-6.5.4-linux-x86_64
3 vim metricbeat.yml
4
5 metricbeat.config.modules:
6   path: ${path.config}/modules.d/*.yml
7   reload.enabled: false
8 setup.template.settings:
9   index.number_of_shards: 1
10  index.codec: best_compression
11 setup.kibana:
12 output.elasticsearch:
13   hosts: ["192.168.40.133:9200", "192.168.40.134:9200", "192.168.40.135:9200"]
14 processors:
15   - add_host_metadata: ~
16   - add_cloud_metadata: ~
17
18 #启动
19 ./metricbeat -e
```

在Elasticsearch中可以看到，系统的一些指标数据已经写入进去了：

查询 1 个分片中用的 1 个, 94 命中, 耗时 0.009 秒

_index	_type	_id	_score ▲	@timestamp	metricset.name	metricset.module	me
metricbeat-6.5.4-2019.01.13	doc	etSLRmgB8SxFt2LBLp1y	1	2019-01-13T09:29:11.306Z	uptime	system	▲
metricbeat-6.5.4-2019.01.13	doc	e9SLRmgB8SxFt2LBLp1y	1	2019-01-13T09:29:11.306Z	fsstat	system	
metricbeat-6.5.4-2019.01.13	doc	fNSLRmgB8SxFt2LBLp1y	1	2019-01-13T09:29:11.306Z	filesystem	system	
metricbeat-6.5.4-2019.01.13	doc	fdSLRmgB8SxFt2LBLp1y	1	2019-01-13T09:29:11.306Z	filesystem	system	
metricbeat-6.5.4-2019.01.13	doc	ftSLRmgB8SxFt2LBLp1y	1	2019-01-13T09:29:11.306Z	filesystem	system	
metricbeat-6.5.4-2019.01.13	doc	f9SLRmgB8SxFt2LBLp1y	1	2019-01-13T09:29:11.306Z	filesystem	system	
metricbeat-6.5.4-2019.01.13	doc	gNSLRmgB8SxFt2LBLp1y	1	2019-01-13T09:29:11.306Z	filesystem	system	
metricbeat-6.5.4-2019.01.13	doc	gdSLRmgB8SxFt2LBLp1y	1	2019-01-13T09:29:11.306Z	filesystem	system	
metricbeat-6.5.4-2019.01.13	doc	gtSLRmgB8SxFt2LBLp1y	1	2019-01-13T09:29:11.306Z	filesystem	system	
metricbeat-6.5.4-2019.01.13	doc	g9SLRmgB8SxFt2LBLp1y	1	2019-01-13T09:29:11.306Z	filesystem	system	
metricbeat-6.5.4-2019.01.13	doc	hNSLRmgB8SxFt2LBLp1y	1	2019-01-13T09:29:11.306Z	filesystem	system	
metricbeat-6.5.4-2019.01.13	doc	hdSLRmgB8SxFt2LBLp1y	1	2019-01-13T09:29:11.307Z	cpu	system	
metricbeat-6.5.4-2019.01.13	doc	htSLRmgB8SxFt2LBLp1y	1	2019-01-13T09:29:11.307Z	load	system	
metricbeat-6.5.4-2019.01.13	doc	h9SLRmgB8SxFt2LBLp1y	1	2019-01-13T09:29:11.307Z	memory	system	
metricbeat-6.5.4-2019.01.13	doc	iNSLRmgB8SxFt2LBLp1y	1	2019-01-13T09:29:11.307Z	network	system	
metricbeat-6.5.4-2019.01.13	doc	idSLRmgB8SxFt2LBLp1y	1	2019-01-13T09:29:11.307Z	network	system	
metricbeat-6.5.4-2019.01.13	doc	itSLRmgB8SxFt2LBLp1y	1	2019-01-13T09:29:11.307Z	network	system	
metricbeat-6.5.4-2019.01.13	doc	i9SLRmgB8SxFt2LBLp1y	1	2019-01-13T09:29:11.307Z	network	system	
metricbeat-6.5.4-2019.01.13	doc	jNSLRmgB8SxFt2LBLp1y	1	2019-01-13T09:29:11.307Z	network	system	
metricbeat-6.5.4-2019.01.13	doc	jdSLRmgB8SxFt2LBLp1y	1	2019-01-13T09:29:11.307Z	network	system	
metricbeat-6.5.4-2019.01.13	doc	jtSLRmgB8SxFt2LBLp1y	1	2019-01-13T09:29:11.307Z	network	system	
metricbeat-6.5.4-2019.01.13	doc	j9SLRmgB8SxFt2LBLp1y	1	2019-01-13T09:29:11.307Z	network	system	
metricbeat-6.5.4-2019.01.13	doc	kNSLRmgB8SxFt2LBLp1y	1	2019-01-13T09:29:11.307Z	network	system	

system module配置：

```

1 root@itcast01:modules.d# cat system.yml
2 # Module: system
3 # Docs: https://www.elastic.co/guide/en/beats/metricbeat/6.5/metricbeat-module-
  system.html
4
5 - module: system
6   period: 10s
7   metricsets:
8     - cpu
9     - load
10    - memory
11    - network
12    - process
13    - process_summary
14    #- core
15    #- diskio
16    #- socket
17  process.include_top_n:
18    by_cpu: 5      # include top 5 processes by CPU
19    by_memory: 5   # include top 5 processes by memory
20
21 - module: system
22   period: 1m
23   metricsets:
24     - filesystem
25     - fsstat
26  processors:
27    - drop_event.when.regexp:
28      system.filesystem.mount_point: '^/(sys|cgroup|proc|dev|etc|host|lib)($|/)'
29
  
```

```
30 - module: system
31   period: 15m
32   metricsets:
33     - uptime
34
35 #- module: system
36 #  period: 5m
37 #  metricsets:
38 #    - raid
39 #  raid.mount_point: '/'
```

5.3、Module

```
1 ./metricbeat modules list #查看列表
2
3 Enabled:
4 system #默认启用
5
6 Disabled:
7 aerospike
8 apache
9 ceph
10 couchbase
11 docker
12 dropwizard
13 elasticsearch
14 envoyproxy
15 etcd
16 golang
17 graphite
18 haproxy
19 http
20 jolokia
21 kafka
22 kibana
23 kubernetes
24 kvm
25 logstash
26 memcached
27 mongodb
28 munin
29 mysql
30 nginx
31 php_fpm
32 postgresql
33 prometheus
34 rabbitmq
35 redis
36 traefik
37 uwsgi
38 vsphere
39 windows
```

5.4、Nginx Module

5.4.1、开启nginx的状态查询

在nginx中，需要开启状态查询，才能查询到指标数据。

```
1  #重新编译nginx
2  ./configure --prefix=/usr/local/nginx --with-http_stub_status_module
3  make
4  make install
5
6  ./nginx -V #查询版本信息
7  nginx version: nginx/1.11.6
8  built by gcc 4.4.7 20120313 (Red Hat 4.4.7-23) (GCC)
9  configure arguments: --prefix=/usr/local/nginx --with-http_stub_status_module
10
11 #配置nginx
12 vim nginx.conf
13 location /nginx-status {
14     stub_status on;
15     access_log off;
16 }
```

测试：



结果说明：

- Active connections：正在处理的活动连接数
- server accepts handled requests
 - 第一个 server 表示Nginx启动到现在共处理了9个连接
 - 第二个 accepts 表示Nginx启动到现在共成功创建 9 次握手
 - 第三个 handled requests 表示总共处理了 21 次请求
 - 请求丢失数 = 握手数 - 连接数，可以看出目前为止没有丢失请求
- Reading: 0 Writing: 1 Waiting: 1
 - Reading：Nginx 读取到客户端的 Header 信息数
 - Writing：Nginx 返回给客户端 Header 信息数
 - Waiting：Nginx 已经处理完正在等候下一次请求指令的驻留链接（开启keep-alive的情况下，这个值等于 Active - (Reading+Writing)）

5.4.2、配置Nginx Module



```
1 #启用redis module
2 ./metricbeat modules enable nginx
3
4 #修改redis module配置
5 vim modules.d/nginx.yml
6
7 # Module: nginx
8 # Docs: https://www.elastic.co/guide/en/beats/metricbeat/6.5/metricbeat-module-
  nginx.html
9
10 - module: nginx
11   #metricsets:
12   #   - stubstatus
13   period: 10s
14
15   # Nginx hosts
16   hosts: ["http://192.168.40.133"]
17
18   # Path to server status. Default server-status
19   server_status_path: "nginx-status"
20
21   #username: "user"
22   #password: "secret"
23
24
25 #启动
26 ./metricbeat -e
```

测试：

```
原始数据
{"_source": {"@timestamp": "2019-03-15T02:46:22.048Z", "metricset": {"rtt": 401, "name": "stubstatus", "module": "nginx", "host": "192.168.40.133"}, "nginx": {"stubstatus": {"active": 1, "accepts": 11, "dropped": 0, "waiting": 0, "handled": 11, "requests": 26, "current": 1, "reading": 0, "writing": 1, "hostname": "192.168.40.133"}}, "host": {"architecture": "x86_64", "name": "node01", "os": {"platform": "centos", "version": "6.9 (Final)", "family": "redhat", "codename": "Final"}, "containerized": true}, "beat": {"name": "node01"}}
```

可以看到，nginx的指标数据已经写入到了Elasticsearch。

更多的Module使用参见官方文档：

<https://www.elastic.co/guide/en/beats/metricbeat/current/metricbeat-modules.html>

6、Kibana



您使用 Elastic Stack 的窗口

通过 Kibana，您能够对 Elasticsearch 中的数据进行可视化并在 Elastic Stack 进行操作，因此您可以在这里解开任何疑问：例如，为何会在凌晨 2:00 收到传呼，雨水会对季度数据造成怎样的影响。



Kibana 是一款开源的数据分析和可视化平台，它是 Elastic Stack 成员之一，设计用于和 Elasticsearch 协作。您可以使用 Kibana 对 Elasticsearch 索引中的数据进行搜索、查看、交互操作。您可以很方便的利用图表、表格及地图对数据进行多元化的分析和呈现。


官网：<https://www.elastic.co/cn/products/kibana>

6.1、配置安装

```
1 #解压安装包
2 tar -xvf kibana-6.5.4-linux-x86_64.tar.gz
3
4 #修改配置文件
5 vim config/kibana.yml
6
7 server.host: "192.168.40.133" #对外暴露服务的地址
8 elasticsearch.url: "http://192.168.40.133:9200" #配置Elasticsearch
9
10 #启动
11 ./bin/kibana
12
13 #通过浏览器进行访问
14 http://192.168.40.133:5601/app/kibana
```

Add Data to Kibana


Use these solutions to quickly turn your data into pre-built dashboards and monitoring systems.



APM

APM automatically collects in-depth performance metrics and errors from inside your applications.


Add APM



Logging

Ingest logs from popular data sources and easily visualize in preconfigured dashboards.


Add log data



Metrics

Collect metrics from the operating system and services running on your servers.

Add metric data



Security analytics

Centralize security events for interactive investigation in ready-to-go visualizations.

Add security events

Add sample data

Load a data set and a Kibana dashboard

Upload data from log file

Import a CSV, NDJSON, or log file

Use Elasticsearch data

Connect to your Elasticsearch index

可以看到kibana页面，并且可以看到提示，导入数据到Kibana。

6.2、功能说明



6.3、数据探索

首先先添加索引信息：

The screenshot shows the Kibana Management / Kibana interface. On the left sidebar, the 'Management' tab is selected. In the main content area, the 'Create index pattern' button is highlighted with a red arrow. Below it, the index pattern 'metricbeat-*' is listed. To the right, a red box highlights the 'metricbeat-*' index pattern. Below this, a green box highlights the 'Time Filter field name: @timestamp' setting. The main content area displays a table of fields for the 'metricbeat-*' index pattern.

Name	Type	Format
@timestamp	date	
_id	string	
_index	string	
_score	number	

即可查看索引数据：

The screenshot shows the Kibana Discover interface. At the top, there are 24 hits. The search bar contains the query 'metricbeat-*'. The left sidebar shows the 'Available fields' list, including '@timestamp', '_id', '_index', '_score', and 'beat.hostname'. The main content area displays a histogram of '@timestamp' per 20 milliseconds. Below the histogram, a table of search results is shown, including fields like '@timestamp', 'host.name', 'host.architecture', 'host.os.family', 'host.os.codename', 'host.os.platform', 'host.os.version', 'host.id', 'metricset.name', 'metricset.module', 'metricset.rtt', 'system.process.cmdline', 'system.process.cpu.total.value', 'system.process.cpu.total.pct', 'system.process.cpu.total.norm.pct', 'system.process.cpu.start_time', and 'system.process.ppid'.

6.4、Metricbeat 仪表盘

可以将Metricbeat的数据在Kibana中展示。

```
1 #修改metricbeat配置
2 setup.kibana:
3   host: "192.168.40.133:5601"
4
5 #安装仪表盘到kibana
6 ./metricbeat setup --dashboards
7
```

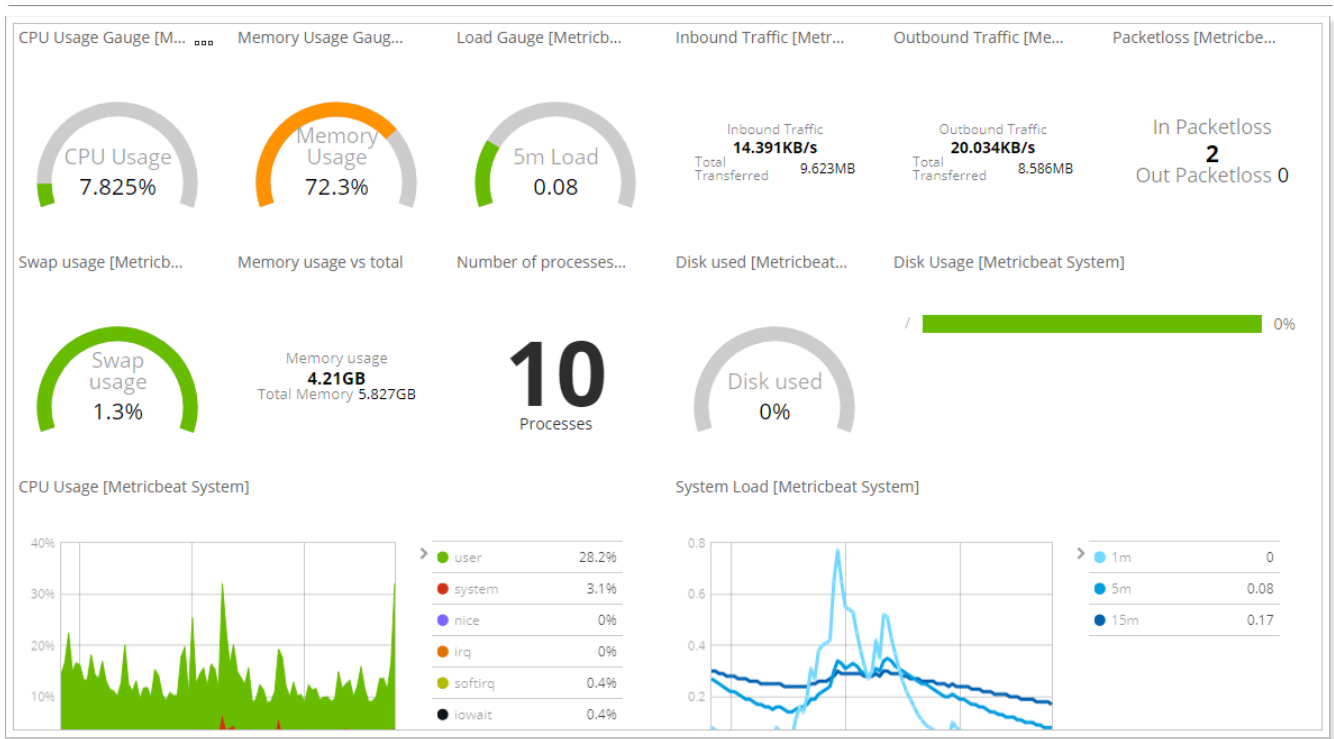
即可在Kibana中看到仪表盘数据：

Dashboards

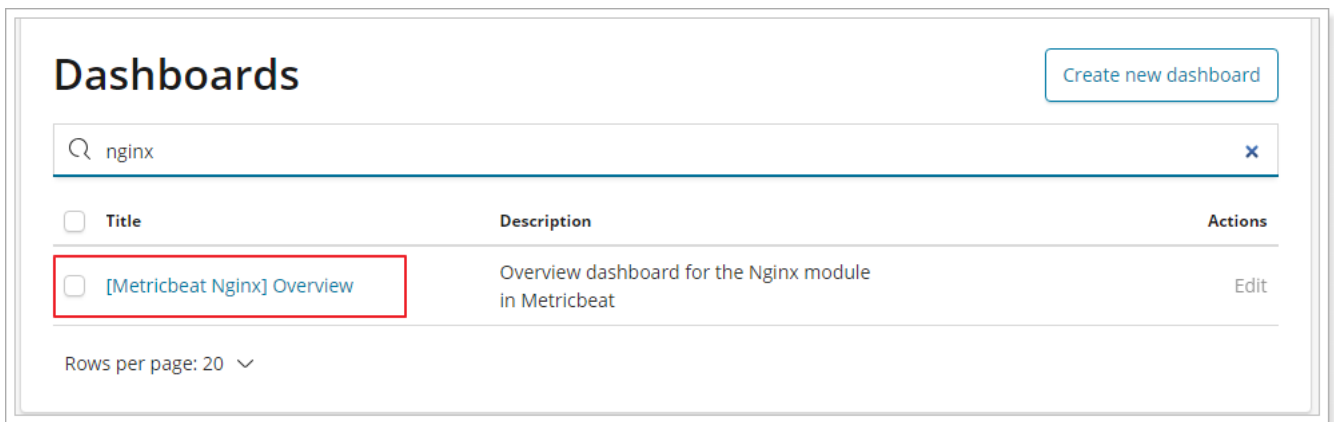
[Create new dashboard](#)

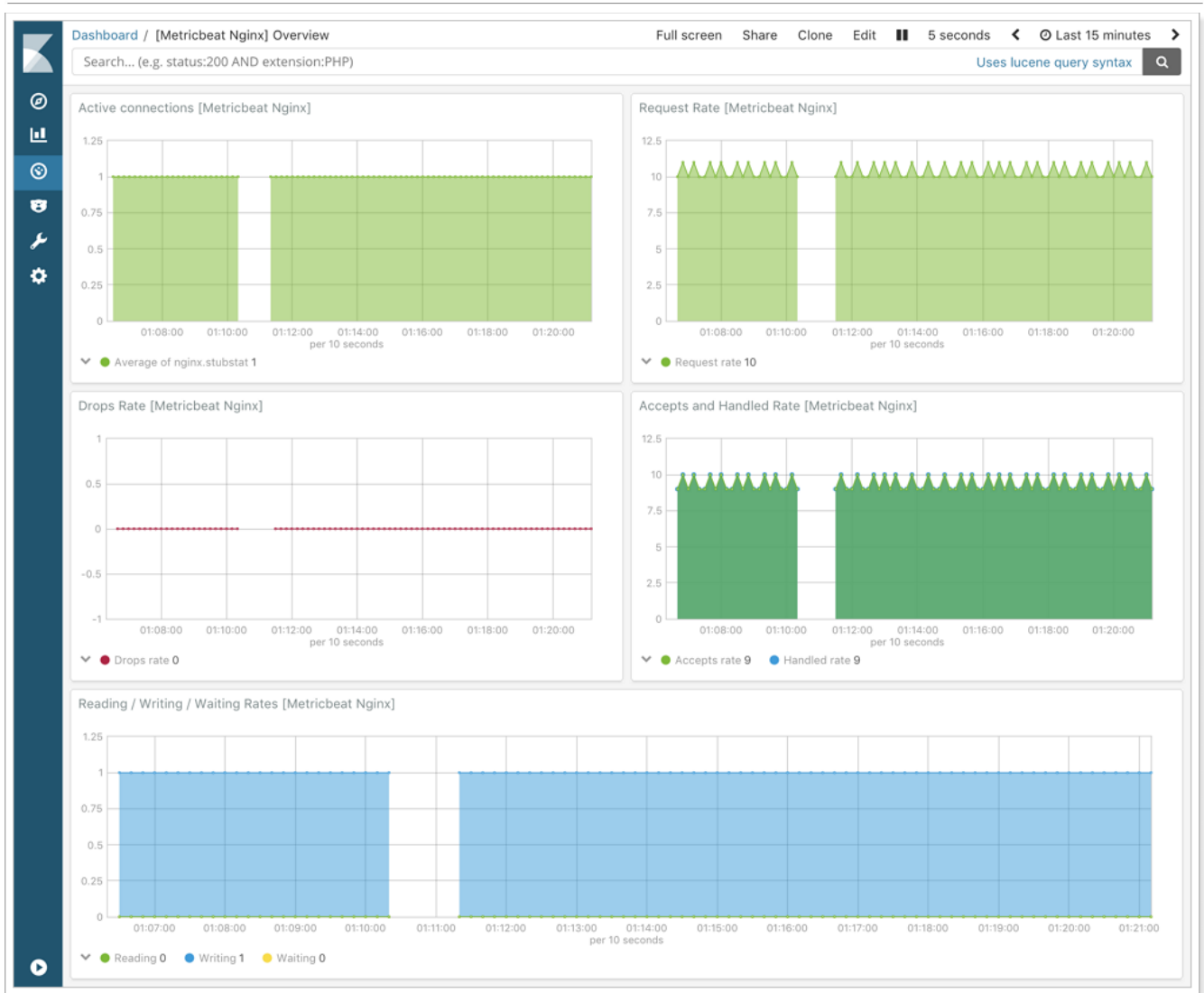
<input type="checkbox"/>	Title	Description	Actions
<input type="checkbox"/>	[Metricbeat Docker] Overview	Overview of docker containers	Edit
<input type="checkbox"/>	[Metricbeat Apache] Overview	Overview of Apache server status	Edit
<input type="checkbox"/>	[Metricbeat System] Containers overview	Overview of container metrics	Edit
<input type="checkbox"/>	[Metricbeat System] Host overview	Overview of host metrics	Edit
<input type="checkbox"/>	[Metricbeat Kafka] Overview	Kafka analysis of topics and consumer groups	Edit
<input type="checkbox"/>	[Metricbeat Golang] Overview	Overview of Go profiling information	Edit
<input type="checkbox"/>	[Metricbeat HAProxy] HTTP backend	HAProxy HTTP backend metrics	Edit
<input type="checkbox"/>	[Metricbeat HAProxy] HTTP server	HAProxy metrics for HTTP mode	Edit
<input type="checkbox"/>	[Metricbeat HAProxy] Backend	HAProxy backend metrics	Edit
<input type="checkbox"/>	[Metricbeat HAProxy] Overview	HAProxy overview	Edit
<input type="checkbox"/>	[Metricbeat HAProxy] HTTP frontend	HAProxy frontend metrics	Edit

查看系统信息：



6.5、Nginx 指标仪表盘





6.6、Nginx 日志仪表盘

```

1  #修改配置文件 vim itcast-nginx.yml
2  filebeat.inputs:
3  #- type: log
4  # enabled: true
5  # paths:
6  #   - /usr/local/nginx/logs/*.log
7  # tags: ["nginx"]
8  setup.template.settings:
9    index.number_of_shards: 3
10 output.elasticsearch:
11   hosts: ["192.168.40.133:9200","192.168.40.134:9200","192.168.40.135:9200"]
12 filebeat.config.modules:
13   path: ${path.config}/modules.d/*.yml
14   reload.enabled: false
15 setup.kibana:
16   host: "192.168.40.133:5601"
17
18
19 #安装仪表盘到kibana

```

```
20 | ./filebeat -c itcast-nginx.yml setup
```

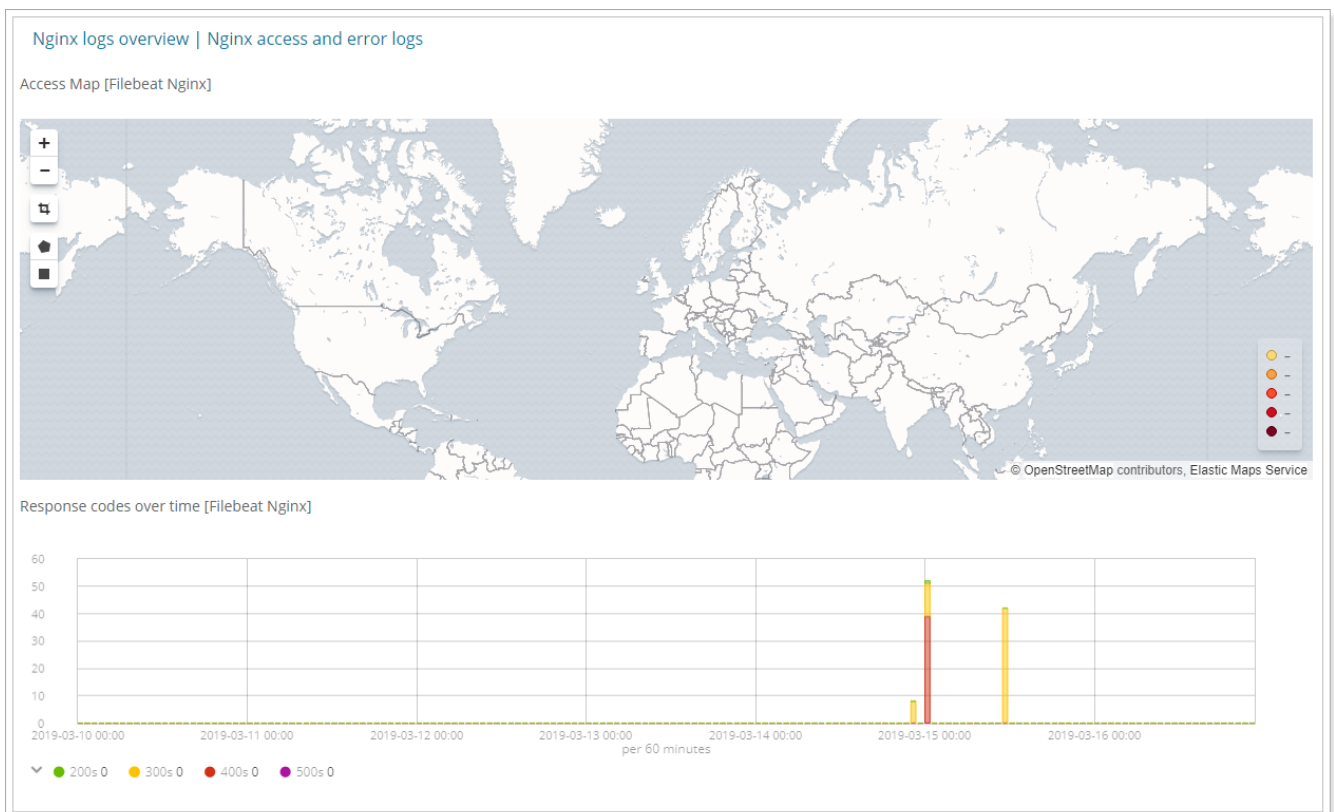
可以看到Nginx的FileBeat的仪表盘了：

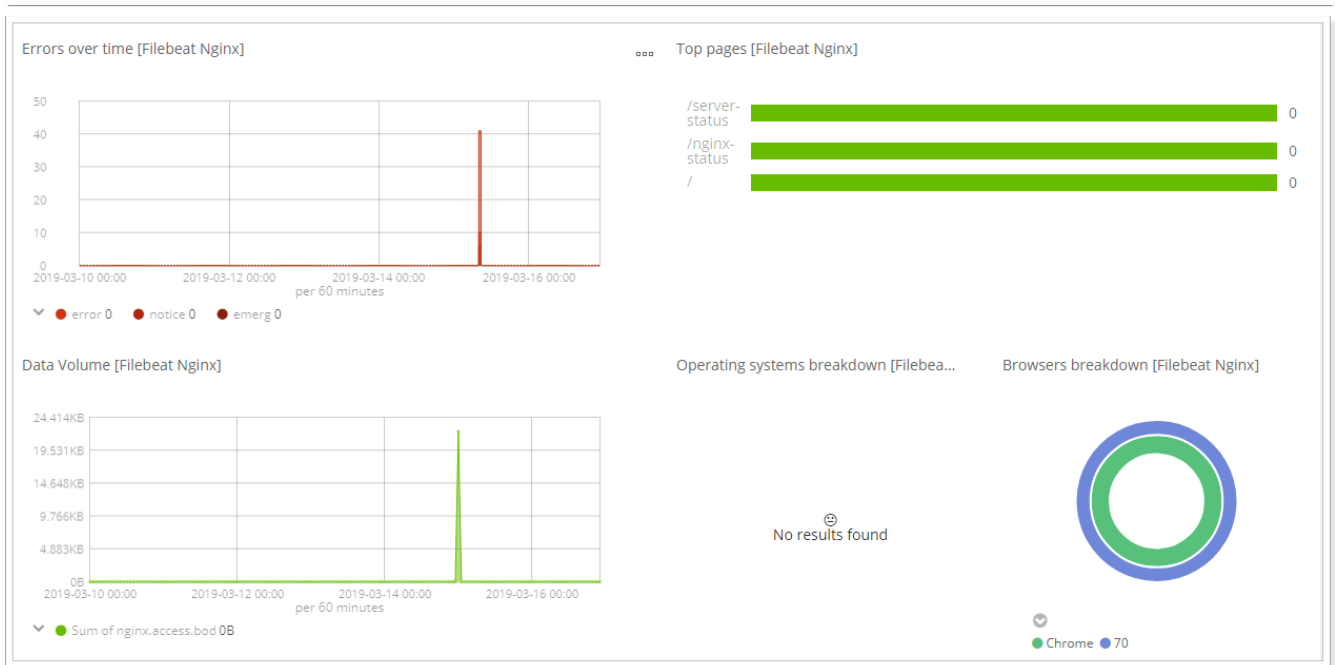
Dashboards

Create new dashboard

<input type="checkbox"/> Title	Description	Actions
<input type="checkbox"/> [Metricbeat Nginx] Overview	Overview dashboard for the Nginx module in Metricbeat	Edit
<input type="checkbox"/> [Filebeat Nginx] Overview	Dashboard for the Filebeat Nginx module	Edit
<input type="checkbox"/> [Filebeat Nginx] [ML] Remote IP URL Explorer	Machine Learning dashboard for the Filebeat Nginx module	Edit
<input type="checkbox"/> [Filebeat Nginx] Access and error logs	Dashboard for the Filebeat Nginx module	Edit
<input type="checkbox"/> [Filebeat Nginx] [ML] Remote IP Count Explorer	Machine learning dashboard, for the Filebeat Nginx module	Edit

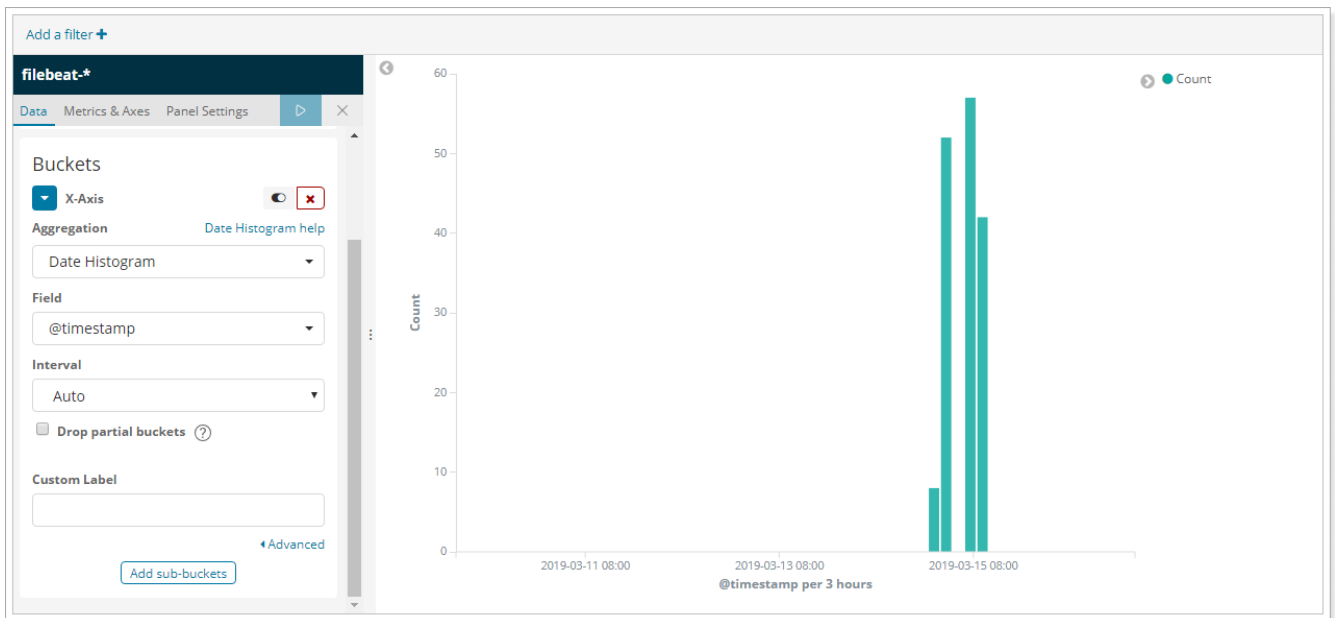
Rows per page: 20 ▾



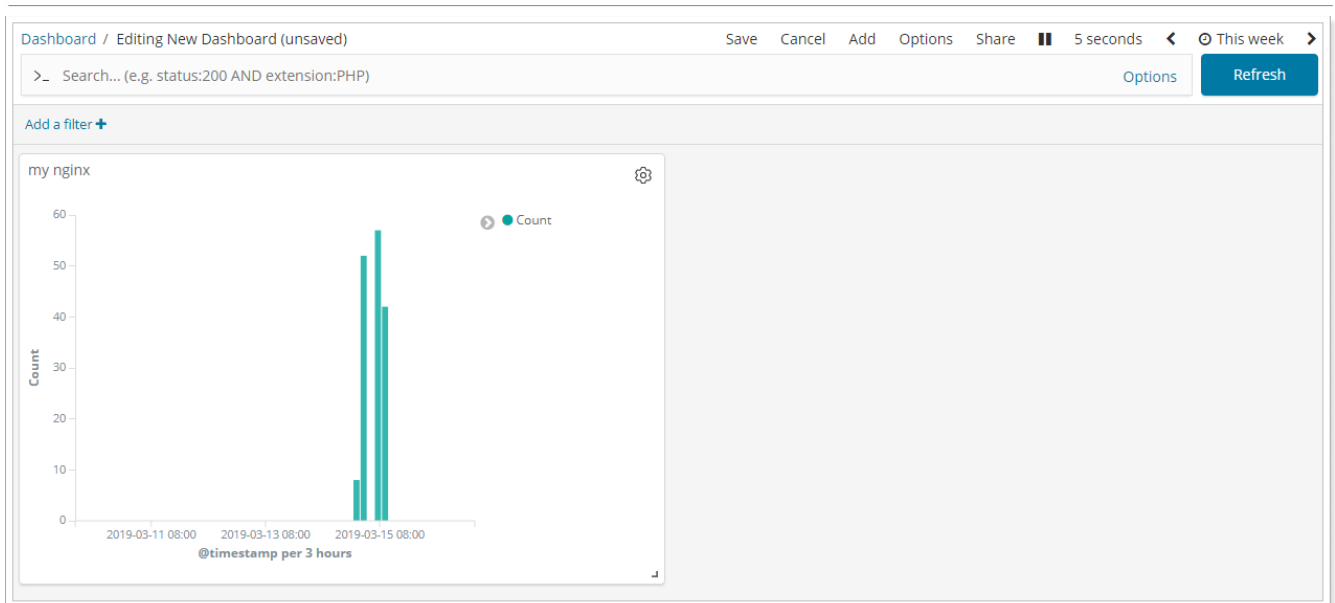


6.7、自定义图表

在Kibana中，也可以进行自定义图表，如制作柱形图：

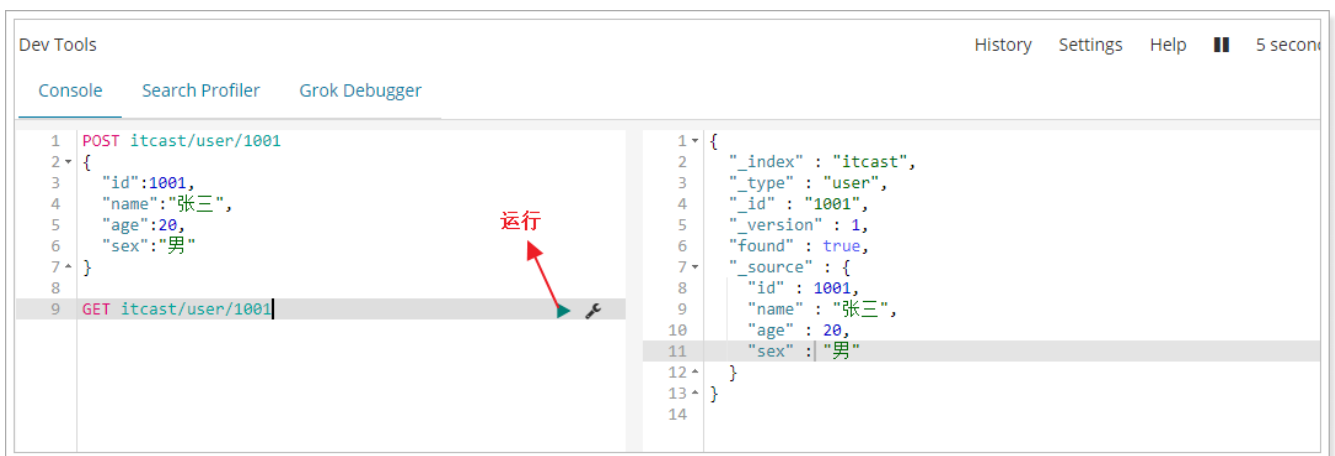


将图表添加到自定义Dashboard中：



6.8、开发者工具

在Kibana中，为开发者的测试提供了便捷的工具使用，如下：

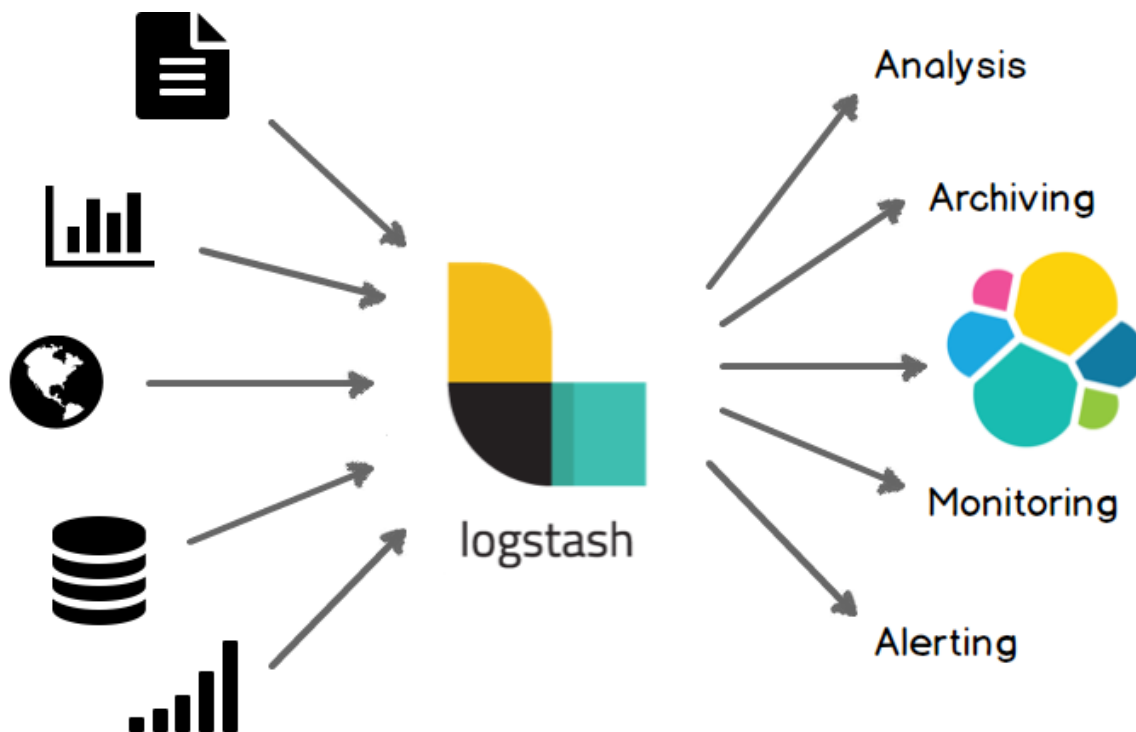


7、Logstash

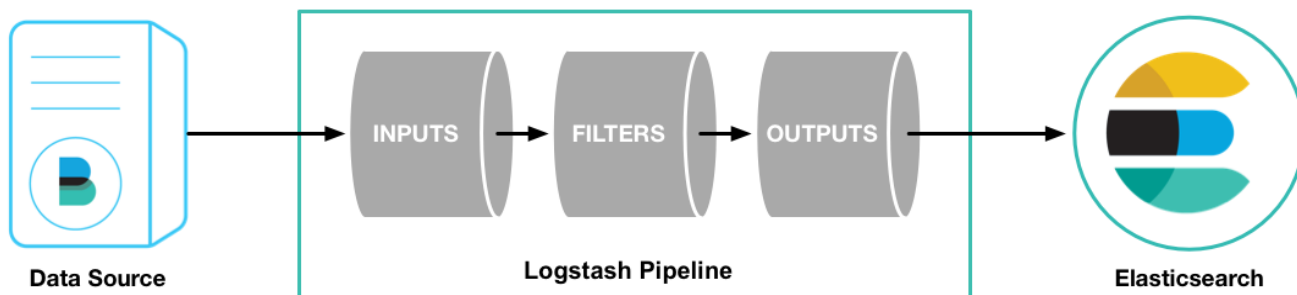
7.1、简介



用途：



7.2、部署安装



```
1 #检查jdk环境，要求jdk1.8+
2 java -version
3
4 #解压安装包
5 tar -xvf logstash-6.5.4.tar.gz
6
7 #第一个logstash示例
8 bin/logstash -e 'input { stdin { } } output { stdout { } }'
```

执行效果如下：

```
[2019-01-14T18:17:44,254][INFO ][logstash.agent
nt {:port=>9600}
hello
{
  "message" => "hello",
  "@timestamp" => 2019-01-14T10:17:51.638Z,
  "host" => "itcast01",
  "@version" => "1"
}
```

7.3、配置详解

Logstash的配置有三部分，如下：

```
1 input { #输入
2     stdin { ... } #标准输入
3 }
4
5 filter { #过滤，对数据进行分割、截取等处理
6     ...
7 }
8
9 output { #输出
10     stdout { ... } #标准输出
11 }
```

7.3.1、输入

- 采集各种样式、大小和来源的数据，数据往往以各种各样的形式，或分散或集中地存在于很多系统中。
- Logstash 支持各种输入选择，可以在同一时间从众多常用来源捕捉事件。能够以连续的流式传输方式，轻松地 从您的日志、指标、Web 应用、数据存储以及各种 AWS 服务采集数据。



7.3.2、过滤

- 实时解析和转换数据
- 数据从源传输到存储库的过程中，Logstash 过滤器能够解析各个事件，识别已命名的字段以构建结构，并将它们转换成通用格式，以便更轻松、更快速地分析和实现商业价值。



7.3.3、输出

Logstash 提供众多输出选择，您可以将数据发送到您要指定的地方，并且能够灵活地解锁众多下游用例。



7.4、读取自定义日志

前面我们通过Filebeat读取了nginx的日志，如果是自定义结构的日志，就需要读取处理后才能使用，所以，这个时候就需要使用Logstash了，因为Logstash有着强大的处理能力，可以应对各种各样的场景。

7.4.1、日志结构

```
1 | 2019-03-15 21:21:21 | ERROR | 读取数据出错 | 参数 : id=1002
```

可以看到，日志中的内容是使用“|”进行分割的，使用，我们在处理的时候，也需要对数据做分割处理。

7.4.2、编写配置文件

```
1 #vim itcast-pipeline.conf
2
3 input {
4   file {
5     path => "/itcast/logstash/logs/app.log"
6     start_position => "beginning"
7   }
8 }
9
10 filter {
11   mutate {
12     split => {"message"=>"|"}
13   }
14 }
15
16 output {
17   stdout { codec => rubydebug }
18 }
```

7.4.3、启动测试

```
1 #启动
2 ./bin/logstash -f ./itcast-pipeline.conf
3
4 #写日志到文件
5 echo "2019-03-15 21:21:21|ERROR|读取数据出错|参数：id=1002" >> app.log
6
7 #输出的结果
8 {
9   "@timestamp" => 2019-03-15T08:44:04.749Z,
10   "path" => "/itcast/logstash/logs/app.log",
11   "@version" => "1",
12   "host" => "node01",
13   "message" => [
14     [0] "2019-03-15 21:21:21",
15     [1] "ERROR",
16     [2] "读取数据出错",
17     [3] "参数：id=1002"
18   ]
19 }
```

可以看到，数据已经被分割了。

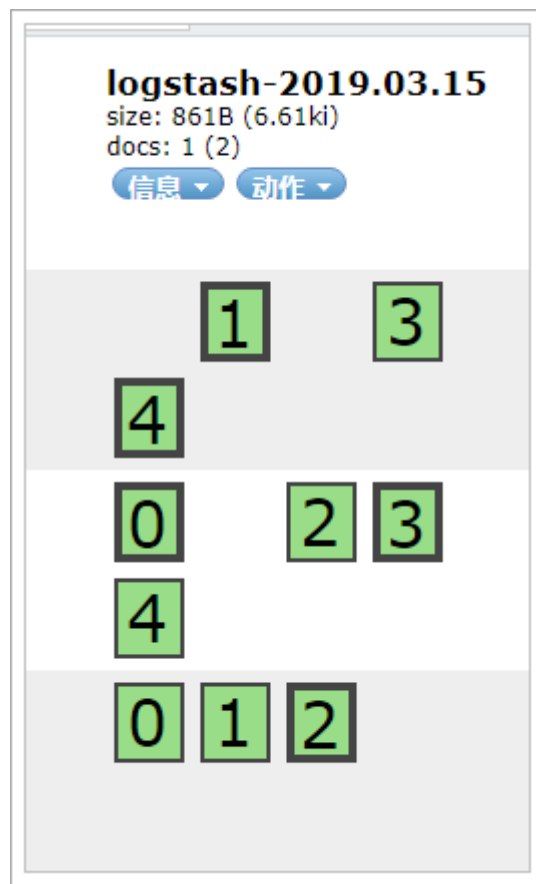
7.4.5、输出到Elasticsearch

```
1 input {
2   file {
3     path => "/itcast/logstash/logs/app.log"
4     #type => "system"
5     start_position => "beginning"
```



```
6   }
7   }
8
9   filter {
10      mutate {
11         split => {"message"=>"|"}
12      }
13   }
14
15   output {
16      elasticsearch {
17         hosts => [ "192.168.40.133:9200", "192.168.40.134:9200", "192.168.40.135:9200" ]
18      }
19   }
20
21
22   #启动
23   ./bin/logstash -f ./itcast-pipeline.conf
24
25   #写入数据
26   echo "2019-03-15 21:21:21|ERROR|读取数据出错|参数：id=1003" >> app.log
```

测试：





查询 5 个分片中的 5 个, 1 命中, 耗时 0.010 秒

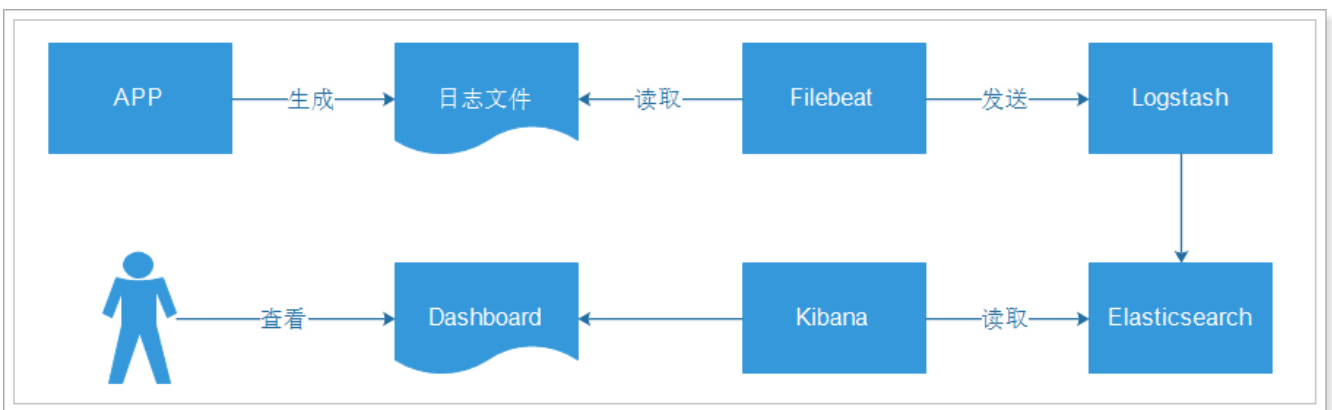
_index	_type	_id	_score	_source	@timestamp	path
logstash	doc	IPKtGkBI5N3X8PeiXSN	1	{ "host": "node01", "@version": "1", "@timestamp": "2019-03-15T08:59:04.675Z", "message": ["2019-03-15 21:21:21"], "ERROR" "读取数据出错" "参数 : id=1003" }	2019-03-15T08:59:04.675Z	/itcast/logstash/logs/app.log

```
{  
  "index": "logstash-2019.03.15",  
  "type": "doc",  
  "id": "IPKtGkBI5N3X8PeiXSN",  
  "version": 1,  
  "score": 1,  
  "_source": {  
    "host": "node01",  
    "@version": "1",  
    "@timestamp": "2019-03-15T08:59:04.675Z",  
    "message": [  
      "2019-03-15 21:21:21"  
    ],  
    "ERROR"  
    "读取数据出错"  
    "参数 : id=1003"  
  },  
  "path": "/itcast/logstash/logs/app.log"  
}
```

8、综合练习

下面我们将前面所学习到的Elasticsearch + Logstash + Beats + Kibana整合起来做一个综合性的练习，目的就是让学生们能够更加深刻的理解Elastic Stack的使用。

8.1、流程说明



- 应用APP生产日志，用来记录用户的操作
 - [INFO] 2019-03-15 22:55:20 [cn.itcast.dashboard.Main] - DAU | 5206 | 使用优惠券 | 2019-03-15 03:37:20
 - [INFO] 2019-03-15 22:55:21 [cn.itcast.dashboard.Main] - DAU | 3880 | 浏览页面 | 2019-03-15 07:25:09
- 通过Filebeat读取日志文件中的内容，并且将内容发送给Logstash，原因是需要对内容做处理
- Logstash接收到内容后，进行处理，如分割操作，然后将内容发送到Elasticsearch中
- Kibana会读取Elasticsearch中的数据，并且在Kibana中进行设计Dashboard，最后进行展示



说明：日志格式、图表、Dashboard都是自定义的。

8.2、APP介绍

APP在生产环境应该是真实的系统，然而，我们现在仅仅的学习，为了简化操作，所以就做数据的模拟生成即可。

业务代码如下：

```
1 package cn.itcast.dashboard;
2
3 import org.apache.commons.lang3.RandomUtils;
4 import org.joda.time.DateTime;
5 import org.slf4j.Logger;
6 import org.slf4j.LoggerFactory;
7 import org.springframework.boot.autoconfigure.SpringBootApplication;
8
9 @SpringBootApplication
10 public class Main {
11
12     private static final Logger LOGGER = LoggerFactory.getLogger(Main.class);
13
14     public static final String[] VISIT = new String[]{"浏览页面", "评论商品", "加入收藏",
15 "加入购物车", "提交订单", "使用优惠券", "领取优惠券", "搜索", "查看订单"};
16
17     public static void main(String[] args) throws Exception {
18         while(true){
19             Long sleep = RandomUtils.nextLong(200, 1000 * 5);
20             Thread.sleep(sleep);
21             Long maxUserId = 9999L;
22             Long userId = RandomUtils.nextLong(1, maxUserId);
23             String visit = VISIT[RandomUtils.nextInt(0, VISIT.length)];
24             DateTime now = new DateTime();
25             int maxHour = now.getHourOfDay();
26             int maxMillis = now.getMinuteOfHour();
27             int maxSeconds = now.getSecondOfMinute();
28             String date = now.plusHours(-(RandomUtils.nextInt(0, maxHour)))
29                 .plusMinutes(-(RandomUtils.nextInt(0, maxMillis)))
30                 .plusSeconds(-(RandomUtils.nextInt(0, maxSeconds)))
31                 .toString("yyyy-MM-dd HH:mm:ss");
32
33             String result = "DAU|" + userId + "|" + visit + "|" + date;
34             LOGGER.info(result);
35         }
36     }
37 }
38
```

运行结果：

```
1 [INFO] 2019-03-15 22:54:42 [cn.itcast.dashboard.Main] - DAU|4645|领取优惠券|2019-03-15
  07:40:29
2 [INFO] 2019-03-15 22:54:44 [cn.itcast.dashboard.Main] - DAU|3482|领取优惠券|2019-03-15
  18:34:04
3 [INFO] 2019-03-15 22:54:48 [cn.itcast.dashboard.Main] - DAU|5607|加入收藏|2019-03-15
  22:44:09
4 [INFO] 2019-03-15 22:54:50 [cn.itcast.dashboard.Main] - DAU|9619|加入收藏|2019-03-15
  21:39:47
5 [INFO] 2019-03-15 22:54:53 [cn.itcast.dashboard.Main] - DAU|7666|加入收藏|2019-03-15
  17:47:18
6 [INFO] 2019-03-15 22:54:54 [cn.itcast.dashboard.Main] - DAU|4871|提交订单|2019-03-15
  02:36:27
7 [INFO] 2019-03-15 22:54:55 [cn.itcast.dashboard.Main] - DAU|7126|加入收藏|2019-03-15
  16:11:06
8 [INFO] 2019-03-15 22:55:00 [cn.itcast.dashboard.Main] - DAU|9606|评论商品|2019-03-15
  02:12:00
9 [INFO] 2019-03-15 22:55:02 [cn.itcast.dashboard.Main] - DAU|7698|查看订单|2019-03-15
  08:17:02
```

代码在资料中可以找到，itcast-dashboard-generate.zip。

部署：

```
1 #打包成jar包，在linux上运行
2 java -jar itcast-dashboard-generate-1.0-SNAPSHOT.jar
3 #运行之后，就可以将日志写入到/itcast/logs/app.log文件中
```

8.3、Filebeat

```
1 #vim itcast-dashboard.yml
2
3 filebeat.inputs:
4 - type: log
5   enabled: true
6   paths:
7     - /itcast/logs/*.log
8 setup.template.settings:
9   index.number_of_shards: 3
10 output.logstash:
11   hosts: ["192.168.40.133:5044"]
12
13
14 #启动
15 ./filebeat -e -c itcast-dashboard.yml
```

8.4、Logstash

```
1 #vim itcast-dashboard.conf
2
3 input {
```




```
4     beats {
5       port => "5044"
6     }
7   }
8
9   filter {
10     mutate {
11       split => {"message"=>"|"}
12     }
13
14
15     mutate {
16       add_field => {
17         "userId" => "%{message[1]}"
18         "visit" => "%{message[2]}"
19         "date" => "%{message[3]}"
20       }
21     }
22     mutate {
23       convert => {
24         "userId" => "integer"
25         "visit" => "string"
26         "date" => "string"
27       }
28     }
29   }
30
31   #output {
32   #     stdout { codec => rubydebug }
33   #}
34
35   output {
36     elasticsearch {
37       hosts => [ "192.168.40.133:9200", "192.168.40.134:9200", "192.168.40.135:9200" ]
38     }
39   }
40
41   #启动
42   ./bin/logstash -f itcast-dashboard.conf
```

8.5、Kibana

启动Kibana：

```
1 #启动
2 ./bin/kibana
3
4 #通过浏览器进行访问
5 http://192.168.40.133:5601/app/kibana
```

添加Logstash索引到Kibana中：

Create index pattern

Kibana uses index patterns to retrieve data from Elasticsearch indices for things like visualizations.

☐ Include system indices

Step 1 of 2: Define index pattern

Index pattern

logstash*

You can use a * as a wildcard in your index pattern.
You can't use spaces or the characters \, /, ?, ", <, >, |.

> Next step

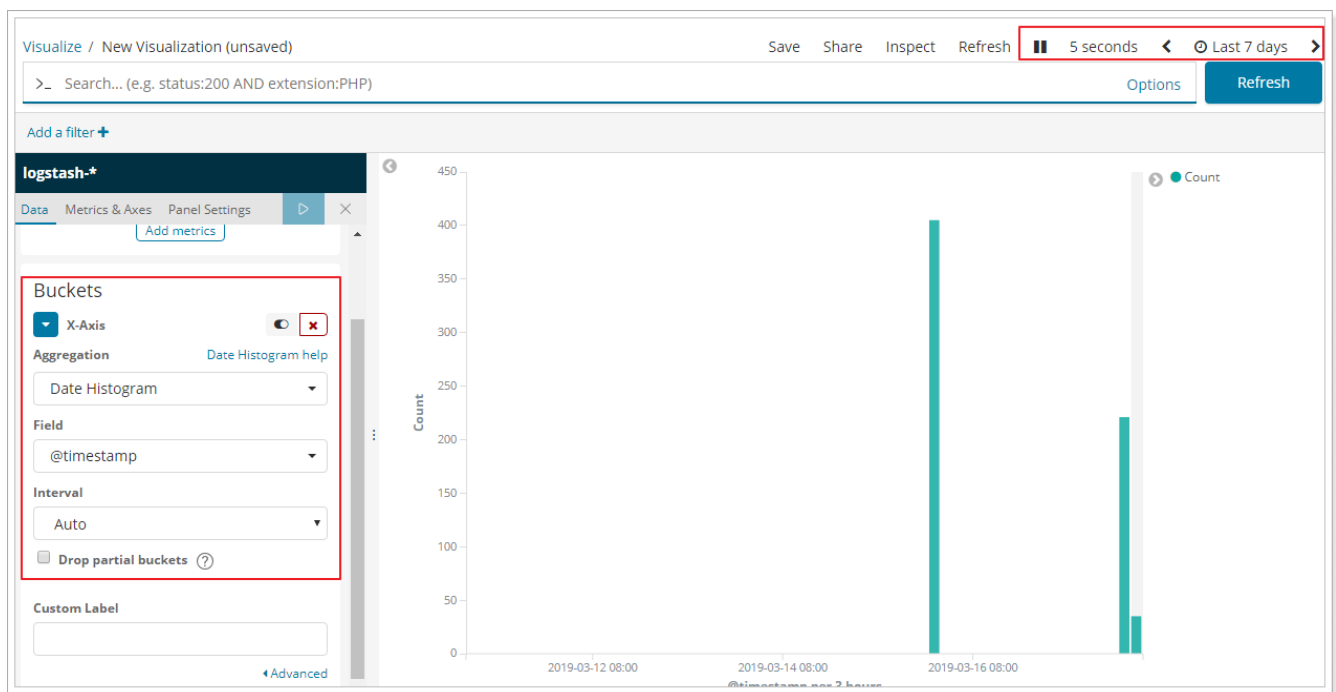
✓ Success! Your index pattern matches 2 indices.

logstash-2019.03.15

logstash-2019.03.17

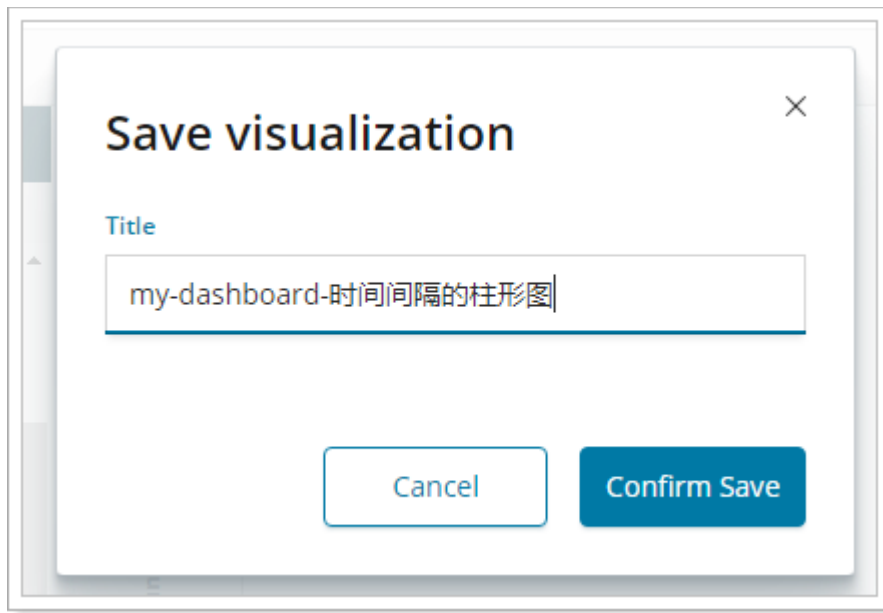
Rows per page: 10

8.5.1、时间间隔的柱形图

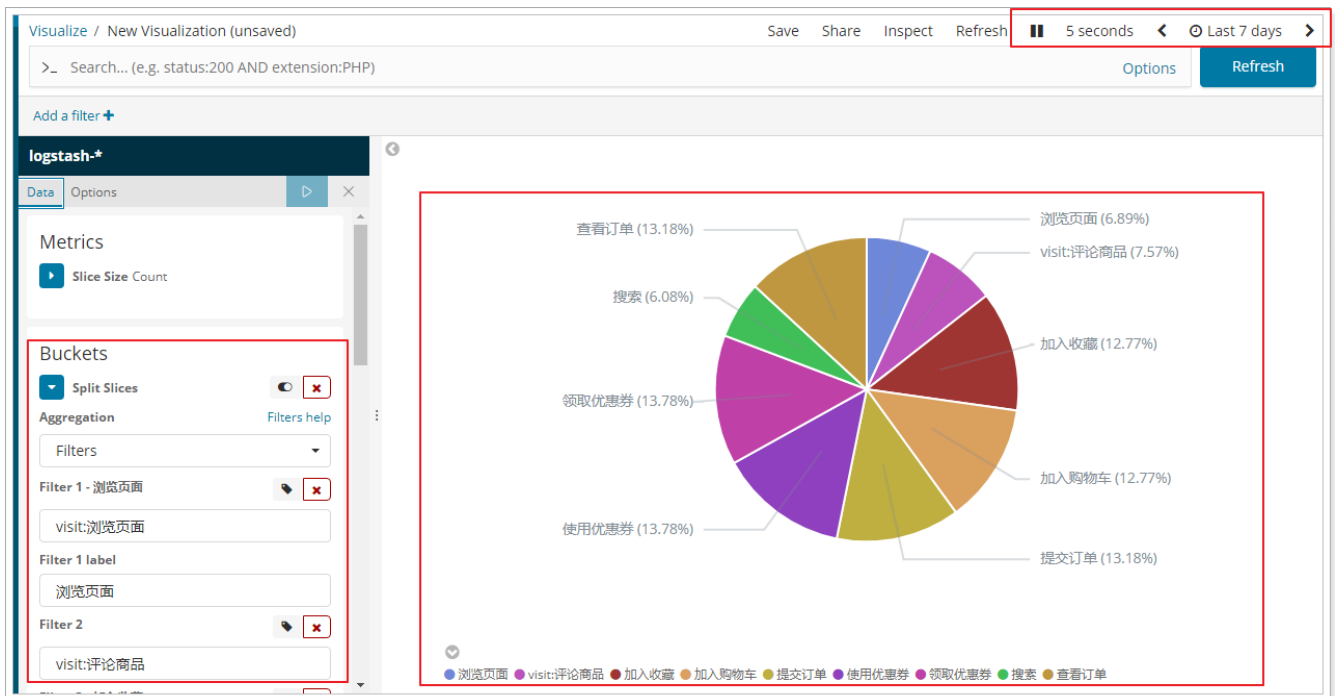


说明：x轴是时间，以天为单位，y轴是count数

保存：（ my-dashboard-时间间隔的柱形图 ）

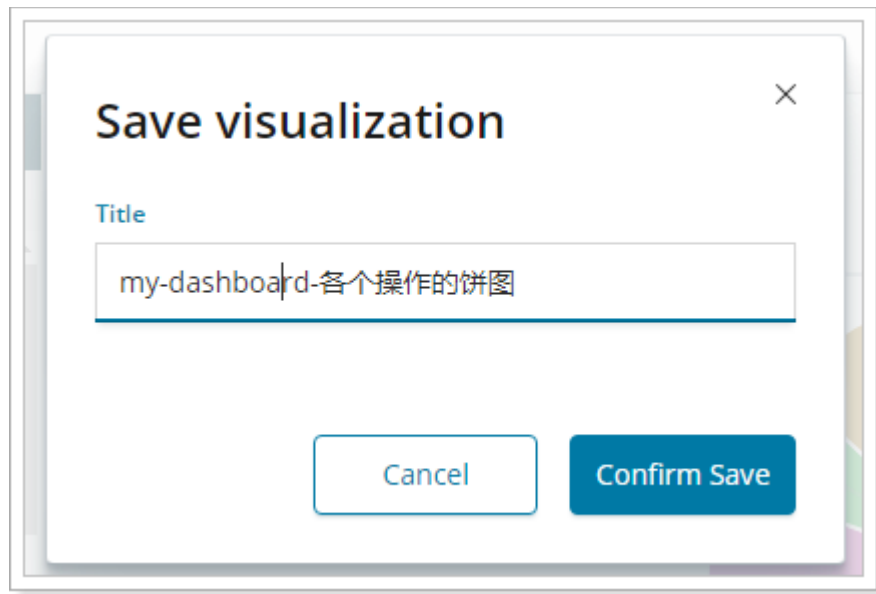


8.5.2、各个操作的饼图分布

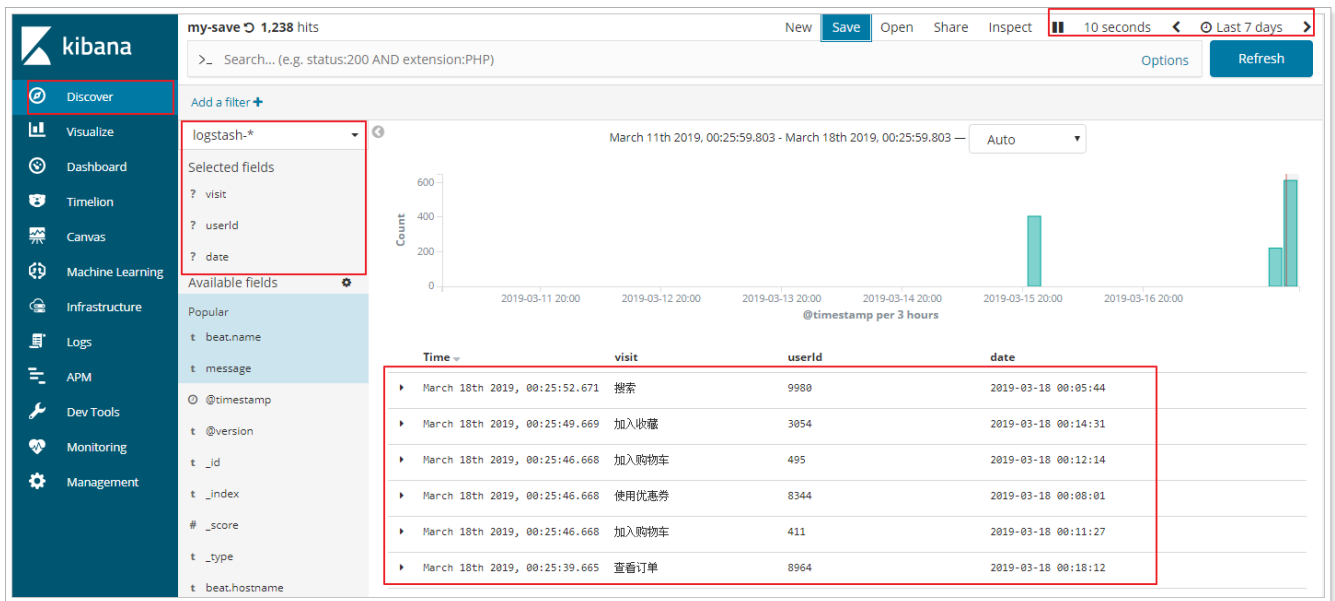


统计各个操作的数量，形成饼图。

保存：（ my-dashboard-各个操作的饼图 ）

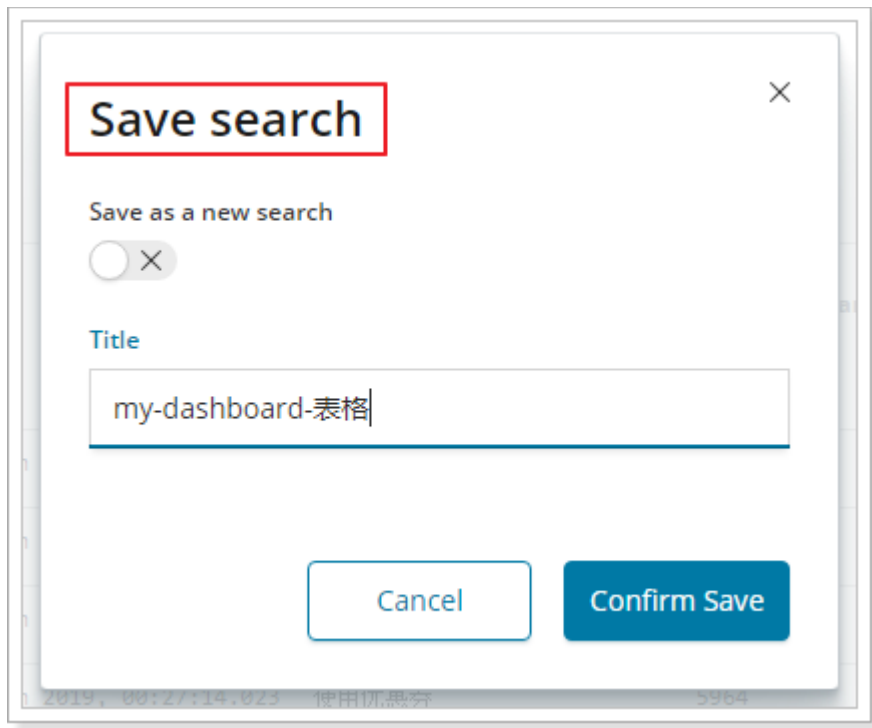


8.5.3、数据表格



在数据探索中进行保存，并且保存，将各个操作的数据以表格的形式展现出来。

保存：（ my-dashboard-表格）



8.5.4、制作Dashboard

