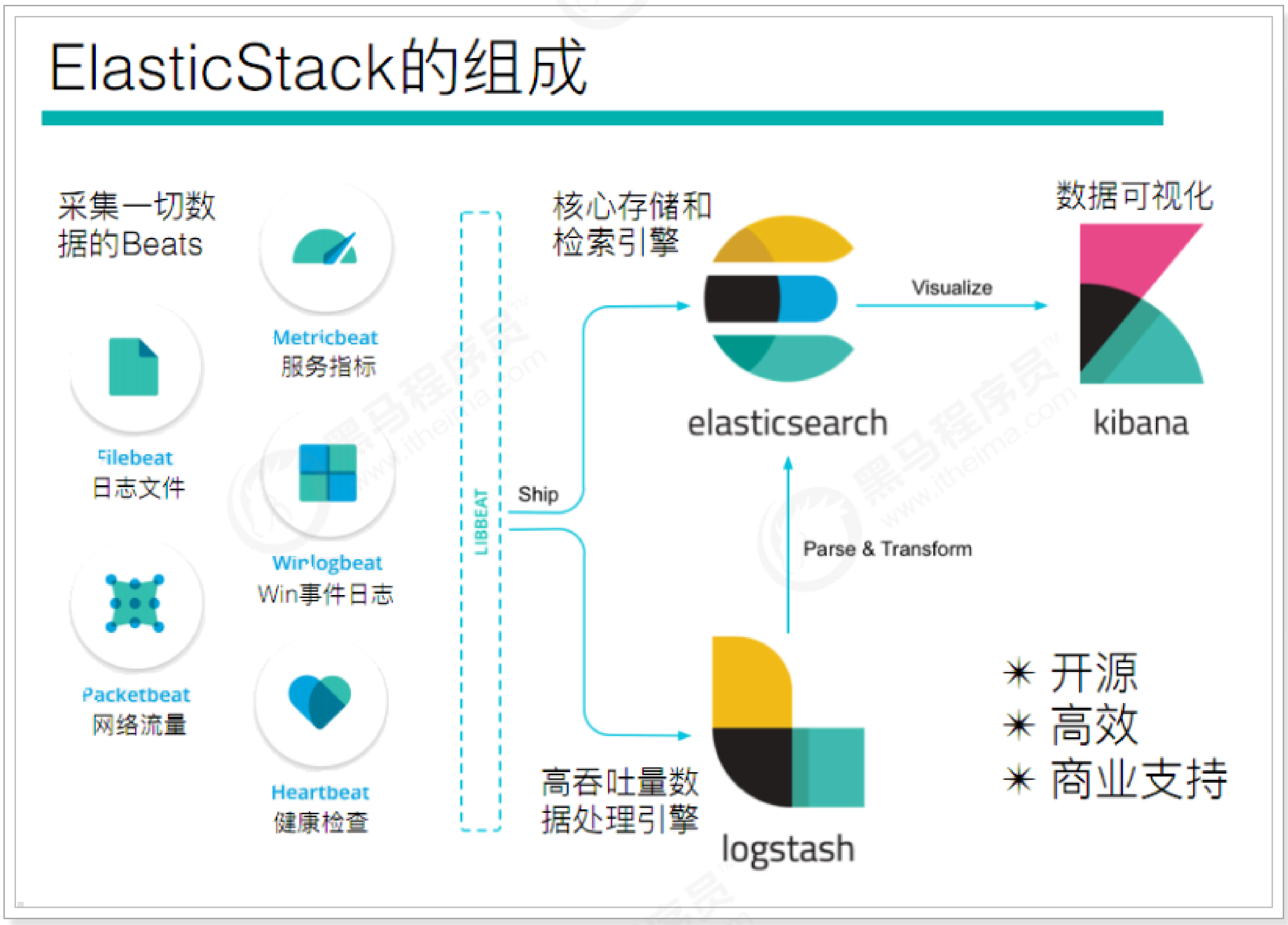
课程介绍

- 了解Beats
- Filebeat入门学习
- Metricbeat入门学习
- Kibana入门学习
- Logstash入门学习

1、Beats 简介

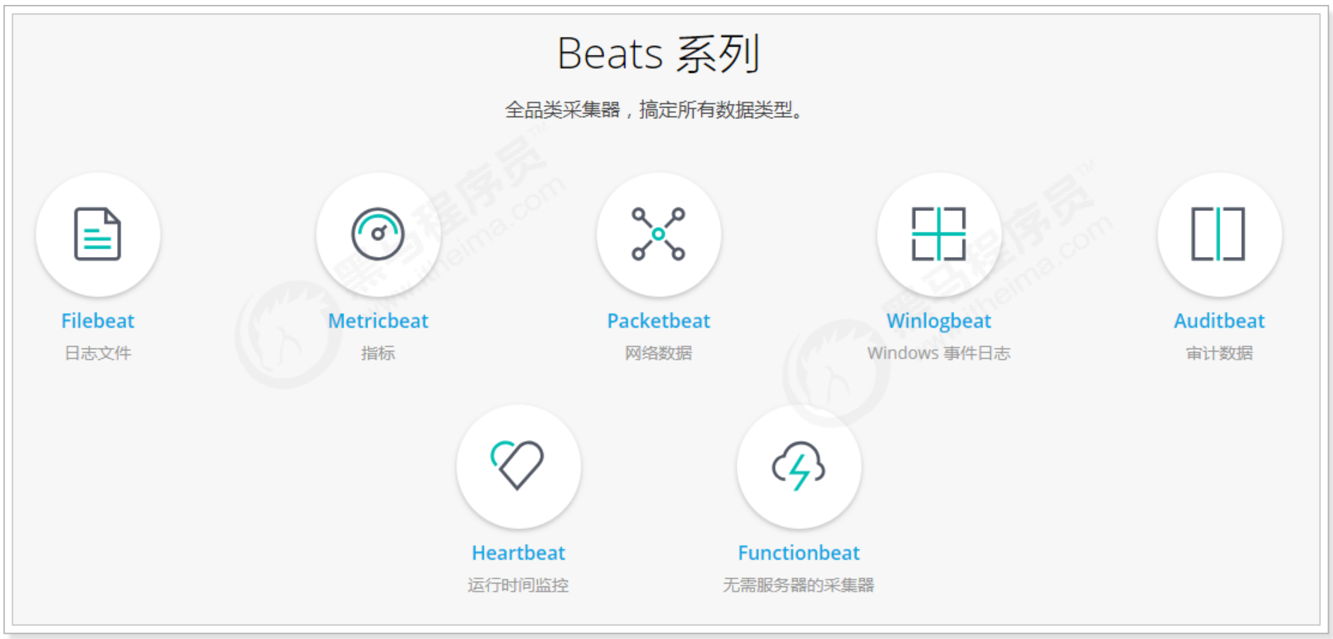


官网：https://www.elastic.co/cn/products/beats

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：

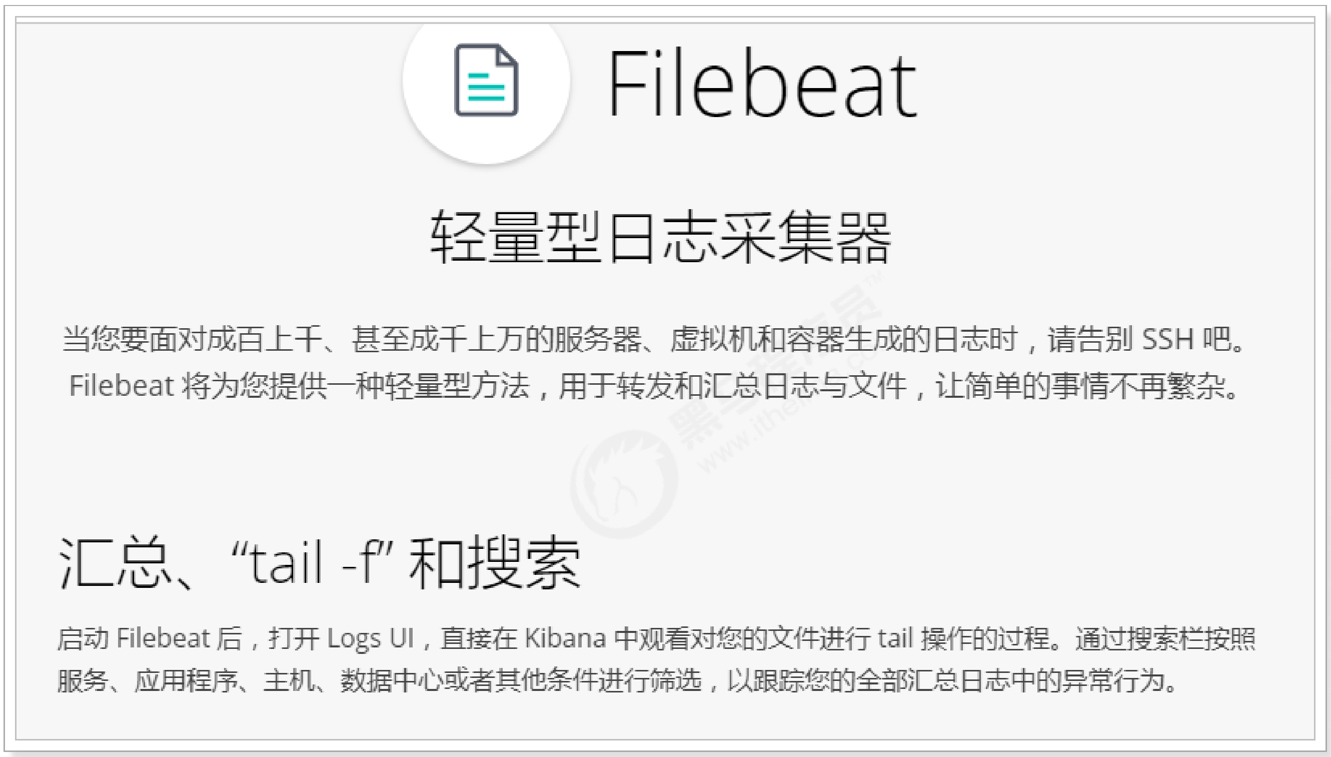


Beats 系列

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

2、Filebeat


Filebeat

轻量型日志采集器

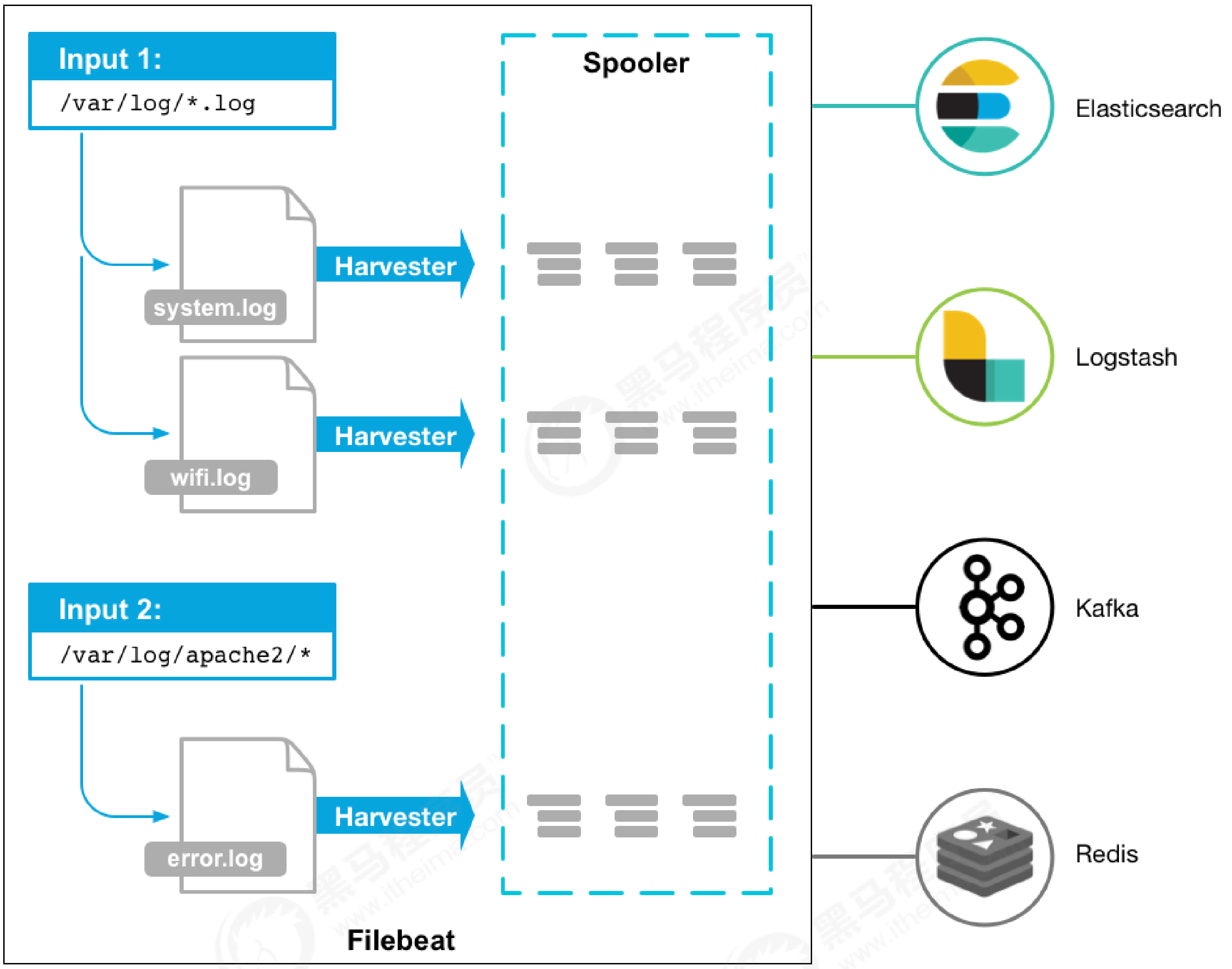
当您要面对成百上千、甚至成千上万的服务器、虚拟机和容器生成的日志时，请告别 SSH 吧。Filebeat 将为您提供一种轻量型方法，用于转发和汇总日志与文件，让简单的事情不再繁杂。

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。

2.1、架构

用于监控、收集服务器日志文件.



2.2、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

```
mkdir /haoke/beats
tar -xvf filebeat-6.5.4-linux-x86_64.tar.gz
cd filebeat-6.5.4-linux-x86_64

#创建如下配置文件 haoke.yml
filebeat.inputs:
- type: stdin
  enabled: true
setup.template.settings:
  index.number_of_shards: 3
output.console:
  pretty: true
  enable: true

#启动filebeat
./filebeat -e -c haoke.yml

#输入hello运行结果如下：
hello
```

```
{
  "@timestamp": "2019-01-12T12:50:03.585Z",
  "@metadata": { #元数据信息
    "beat": "filebeat",
    "type": "doc",
    "version": "6.5.4"
  },
  "source": "",
  "offset": 0,
  "message": "hello",   #输入的内容
  "prospector": { #标准输入勘探器
    "type": "stdin"
  },
  "input": {   #控制台标准输入
    "type": "stdin"
  },
  "beat": { #beat版本以及主机信息
    "name": "itcast01",
    "hostname": "itcast01",
    "version": "6.5.4"
  },
  "host": {
    "name": "itcast01"
  }
}

```

2.3、读取文件

```
#配置读取文件项 haoke-log.yml

filebeat.inputs:
- type: log
  enabled: true
  paths:
    - /haoke/beats/logs/*.log
setup.template.settings:
  index.number_of_shards: 3
output.console:
  pretty: true
  enable: true

#启动filebeat
./filebeat -e -c haoke-log.yml

#/haoke/beats/logs下创建a.log文件，并输入如下内容
hello
world

#观察filebeat输出
{
  "@timestamp": "2019-01-12T14:16:10.192Z",
  "@metadata": {
    "beat": "filebeat",
    "type": "doc",
    "version": "6.5.4"
  },
  "host": {
    "name": "itcast01"
  },
  "source": "/haoke/beats/logs/a.log",
  "offset": 0,
  "message": "hello",
  "prospector": {
    "type": "log"
  },
  "input": {
    "type": "log"
  },
  "beat": {
    "version": "6.5.4",
    "name": "itcast01",
    "hostname": "itcast01"
  }
}
{
  "@timestamp": "2019-01-12T14:16:10.192Z",
  "@metadata": {
    "beat": "filebeat",
    "type": "doc",
    "version": "6.5.4"
  },
  "prospector": {
    "type": "log"
  },
  "input": {
    "type": "log"
  },
  "beat": {
    "version": "6.5.4",
    "name": "itcast01",
    "hostname": "itcast01"
  },
  "host": {
    "name": "itcast01"
  },
  "source": "/haoke/beats/logs/a.log",
  "offset": 6,
  "message": "world"
}

```

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。

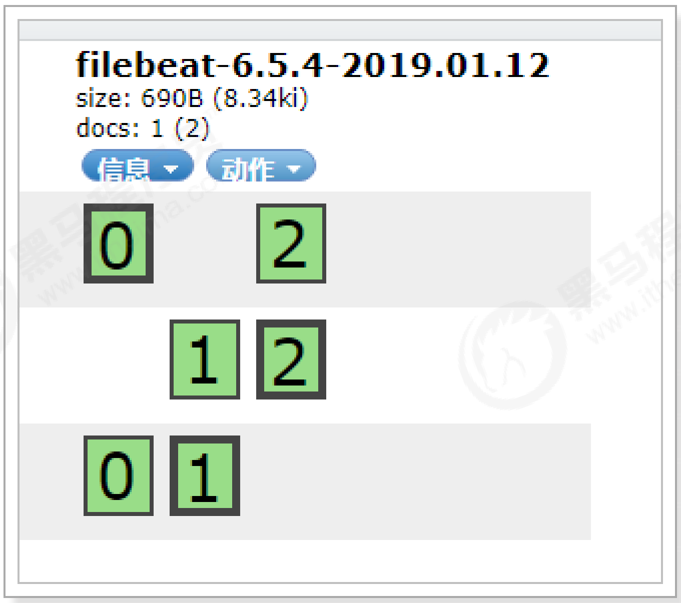
2.4、自定义字段

```
#配置读取文件项 haoke-log.yml

filebeat.inputs:
- type: log
  enabled: true
  paths:
    - /haoke/beats/logs/*.log
  tags: ["web"]   #添加自定义tag，便于后续的处理
  fields:  #添加自定义字段
    from: haoke-im
  fields_under_root: true #true为添加到根节点，false为添加到子节点中
setup.template.settings:
  index.number_of_shards: 3
output.console:
  pretty: true
  enable: true

#启动filebeat
./filebeat -e -c haoke-log.yml

#/haoke/beats/logs下创建a.log文件，并输入如下内容
123

#执行效果
{
  "@timestamp": "2019-01-12T14:37:19.845Z",
  "@metadata": {
    "beat": "filebeat",
    "type": "doc",
    "version": "6.5.4"
  },
  "offset": 0,
  "tags": [
    "haoke-im"
  ],
  "prospector": {
    "type": "log"
  },
  "beat": {
    "name": "itcast01",
    "hostname": "itcast01",
    "version": "6.5.4"
  },
  "host": {
    "name": "itcast01"
  },
  "source": "/haoke/beats/logs/a.log",
  "message": "123",
  "input": {
    "type": "log"
  },
  "from": "haoke-im"
}
```

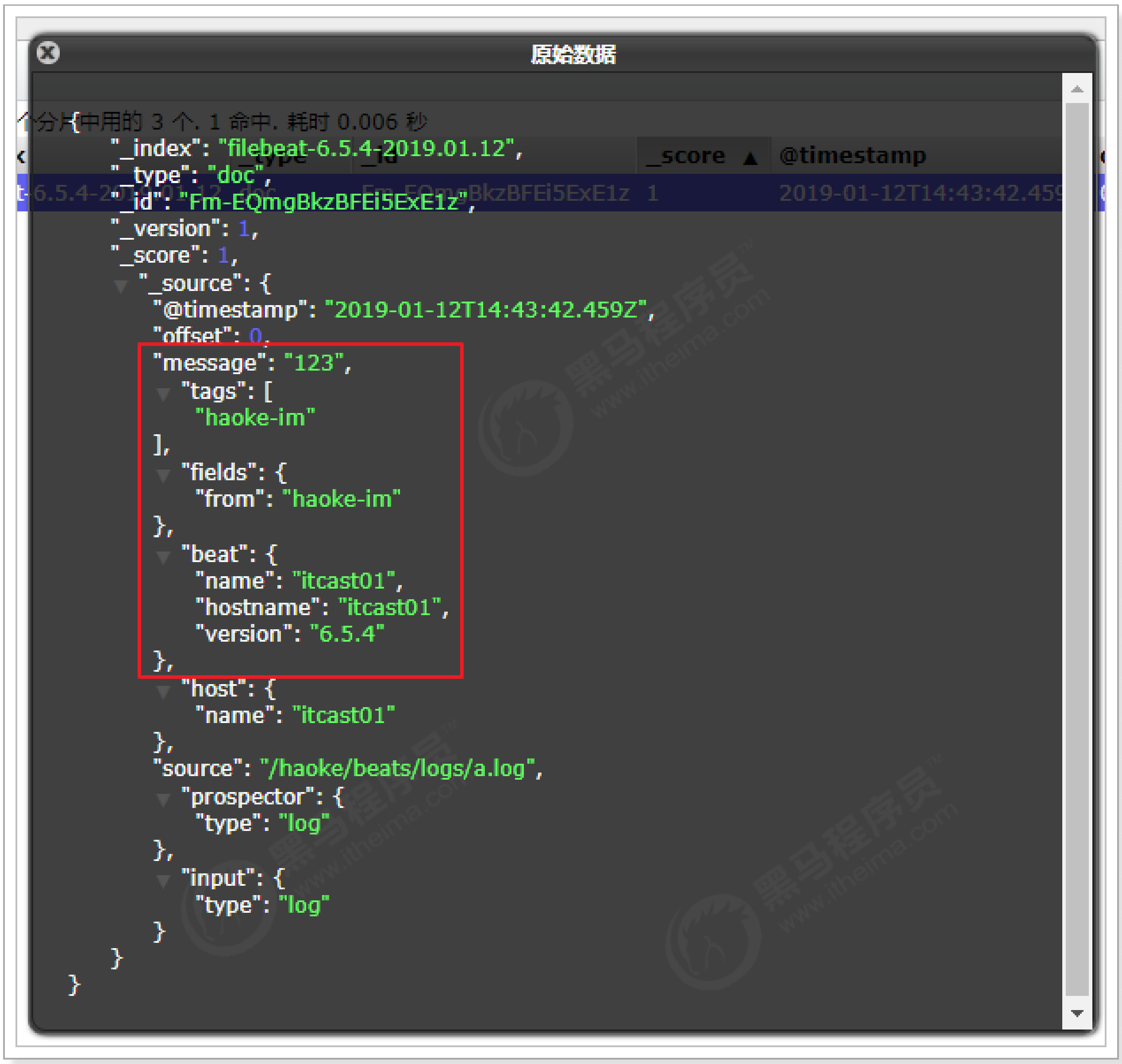
2.5、输出到Elasticsearch

```yaml
# haoke-log.yml
filebeat.inputs:
- type: log
  enabled: true
  paths:
    - /haoke/beats/logs/*.log
  tags: ["haoke-im"]
  fields:
    from: haoke-im
  fields_under_root: false
setup.template.settings:
  index.number_of_shards: 3 #指定索引的分区数
output.elasticsearch: #指定ES的配置
  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：


filebeat-6.5.4-2019.01.12
size: 690B (8.34ki)
docs: 1 (2)

查看数据：



```
"_index": "filebeat-6.5.4-2019.01.12",
"_type": "doc",
"_id": "Fm-EQmgBkzBFEi5ExE1z",
"_version": 1,
"_score": 1,
"_source": {
  "@timestamp": "2019-01-12T14:43:42.459Z",
  "offset": 0,
  "message": "123",
    "tags": [
      "haoke-im"
  ],
    "fields": {
      "from": "haoke-im"
  },
    "beat": {
      "name": "itcast01",
      "hostname": "itcast01",
      "version": "6.5.4"
  },
    "host": {
      "name": "itcast01"
  },
  "source": "/haoke/beats/logs/a.log",
    "prospector": {
      "type": "log"
  },
    "input": {
      "type": "log"
  }
  }
}
```

2.6、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：

```
./filebeat -e -c haoke.yml
./filebeat -e -c haoke.yml -d "publish"

#参数说明
 -e: 输出到标准输出，默认输出到syslog和logs下
 -c: 指定配置文件
 -d: 输出debug信息

#测试： ./filebeat -e -c haoke-log.yml -d "publish"
 DEBUG    [publish]       pipeline/processor.go:308       Publish event: {
  "@timestamp": "2019-01-12T15:03:50.820Z",
  "@metadata": {
    "beat": "filebeat",
    "type": "doc",
    "version": "6.5.4"
  },
  "offset": 0,
  "tags": [
    "haoke-im"
  ],
  "input": {
    "type": "log"
  },
  "prospector": {
    "type": "log"
  },
  "beat": {
    "name": "itcast01",
    "hostname": "itcast01",
    "version": "6.5.4"
  },
  "source": "/haoke/beats/logs/a.log",
  "fields": {
    "from": "haoke-im"
  },
  "host": {
    "name": "itcast01"
  },
  "message": "456"
}

```

2.7、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：

```
./filebeat modules list

Enabled:

Disabled:
apache2
auditd
elasticsearch
haproxy
icinga
iis
kafka
kibana
logstash
mongodb
mysql
nginx
osquery
postgresql
redis
suricata
system
traefik

```

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：

```
./filebeat modules enable redis #启动
./filebeat modules disable redis #禁用

Enabled:
redis

Disabled:
apache2
auditd
elasticsearch
haproxy
icinga
iis
kafka
kibana
logstash
mongodb
mysql
nginx
osquery
postgresql
suricata
system
traefik
```

可以发现，redis的module已经被启用。

2.7.1、redis module

module目录：

```
.
├── log  #日志
│   ├── config
│   │   └── log.yml
│   ├── ingest
│   │   └── pipeline.json
│   └── manifest.yml
├── module.yml
└── slowlog #慢查询日志
    ├── config
    │   └── slowlog.yml
    ├── ingest
    │   └── pipeline.json
    └── manifest.yml
```

2.7.2、redis module 配置

```
cd modules.d/
vim redis.yml

- module: redis
  # Main logs
  log:
    enabled: true

    # Set custom paths for the log files. If left empty,
    # Filebeat will choose the paths depending on your OS.
    var.paths: ["/data/redis-data/node01/*.log"]

  # Slow logs, retrieved via the Redis API (SLOWLOG)
  slowlog:
    enabled: false

    # The Redis hosts to connect to.
    #var.hosts: ["localhost:6379"]

    # Optional, the password to use when connecting to Redis.
    #var.password:
```

2.7.3、修改redis的docker容器

redis默认情况下，是不会输出日志的，需要进行配置，前面我们使用的容器都没有配置日志输出，下面需要配置一下。

```
docker create --name redis-node01 -v /data/redis-data/node01:/data -p 6379:6379 redis:5.0.2 --cluster-enabled yes --cluster-config-file nodes-node-01.conf --loglevel debug --logfile nodes-node-01.log

docker create --name redis-node02 -v /data/redis-data/node02:/data -p 6380:6379 redis:5.0.2 --cluster-enabled yes --cluster-config-file nodes-node-02.conf --loglevel debug --logfile nodes-node-02.log

docker create --name redis-node03 -v /data/redis-data/node03:/data -p 6381:6379 redis:5.0.2 --cluster-enabled yes --cluster-config-file nodes-node-03.conf --loglevel debug --logfile nodes-node-03.log
```

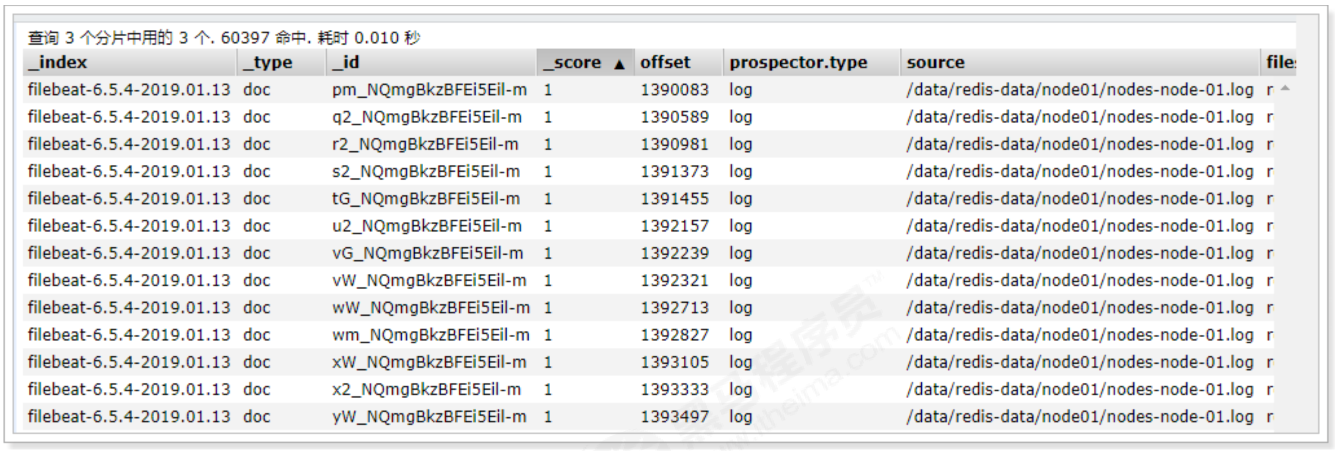
> loglevel 日志等级分为：debug、verbose、notice、warning
>
> 其中，debug 会有大量信息，对开发、测试有用；
>
> verbose 等于log4j 中的info，有很多信息，但是不会像debug那样乱；
>
> notice 一般信息；
>
> warning 只有非常重要/关键的消息被记录。

2.7.4、配置filebeat

```
#vim haoke-redis.yml

filebeat.inputs:
- type: log
  enabled: true
  paths:
      - /haoke/log/*.log
setup.template.settings:
  index.number_of_shards: 3
output.console:
  pretty: true
  enable: true
filebeat.config.modules:
  path: ${path.config}/modules.d/*.yml
  reload.enabled: false

```

2.7.5、测试

```
./filebeat -e -c haoke-redis.yml --modules redis
```



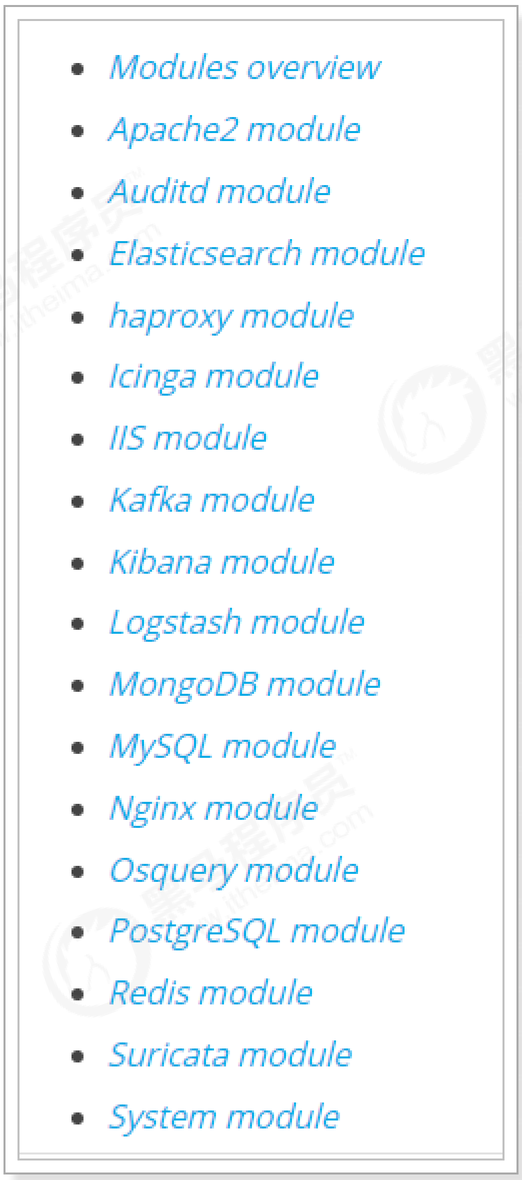
查询 3 个分片中用的 3 个. 60397 命中. 耗时 0.010 秒

_index	_type	_id	_score ▲	offset	prospector.type	source	file
filebeat-6.5.4-2019.01.13	doc	pm_NQmgBkzBFEi5Eil-m	1	1390083	log	/data/redis-data/node01/nodes-node-01.log	r
filebeat-6.5.4-2019.01.13	doc	q2_NQmgBkzBFEi5Eil-m	1	1390589	log	/data/redis-data/node01/nodes-node-01.log	r
filebeat-6.5.4-2019.01.13	doc	r2_NQmgBkzBFEi5Eil-m	1	1390981	log	/data/redis-data/node01/nodes-node-01.log	r
filebeat-6.5.4-2019.01.13	doc	s2_NQmgBkzBFEi5Eil-m	1	1391373	log	/data/redis-data/node01/nodes-node-01.log	r
filebeat-6.5.4-2019.01.13	doc	tG_NQmgBkzBFEi5Eil-m	1	1391455	log	/data/redis-data/node01/nodes-node-01.log	r
filebeat-6.5.4-2019.01.13	doc	u2_NQmgBkzBFEi5Eil-m	1	1392157	log	/data/redis-data/node01/nodes-node-01.log	r
filebeat-6.5.4-2019.01.13	doc	vG_NQmgBkzBFEi5Eil-m	1	1392239	log	/data/redis-data/node01/nodes-node-01.log	r
filebeat-6.5.4-2019.01.13	doc	vW_NQmgBkzBFEi5Eil-m	1	1392321	log	/data/redis-data/node01/nodes-node-01.log	r
filebeat-6.5.4-2019.01.13	doc	wW_NQmgBkzBFEi5Eil-m	1	1392713	log	/data/redis-data/node01/nodes-node-01.log	r
filebeat-6.5.4-2019.01.13	doc	wm_NQmgBkzBFEi5Eil-m	1	1392827	log	/data/redis-data/node01/nodes-node-01.log	r
filebeat-6.5.4-2019.01.13	doc	xW_NQmgBkzBFEi5Eil-m	1	1393105	log	/data/redis-data/node01/nodes-node-01.log	r
filebeat-6.5.4-2019.01.13	doc	x2_NQmgBkzBFEi5Eil-m	1	1393333	log	/data/redis-data/node01/nodes-node-01.log	r
filebeat-6.5.4-2019.01.13	doc	yW_NQmgBkzBFEi5Eil-m	1	1393497	log	/data/redis-data/node01/nodes-node-01.log	r

测试发现，数据已经写入到了Elasticsearch中。

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

3、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。

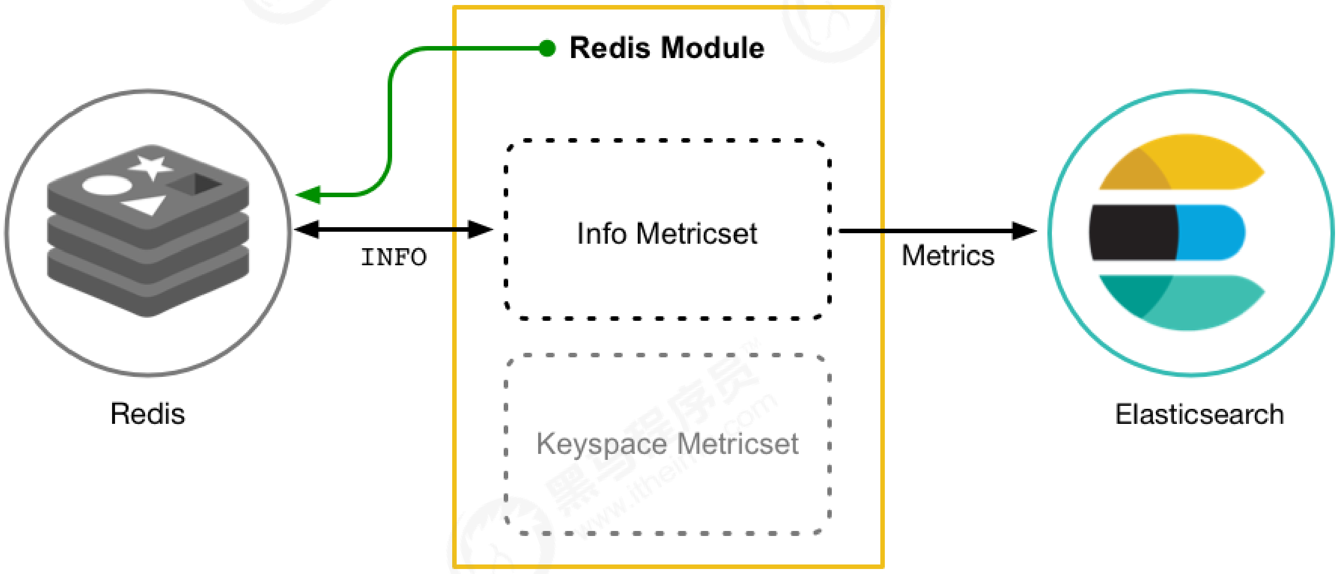
- 定期收集操作系统或应用服务的指标数据
- 存储到Elasticsearch中，进行实时分析

3.1、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、操作系统等；
- Metricset
    - 收集指标的集合，如：cpu、memory、network等；

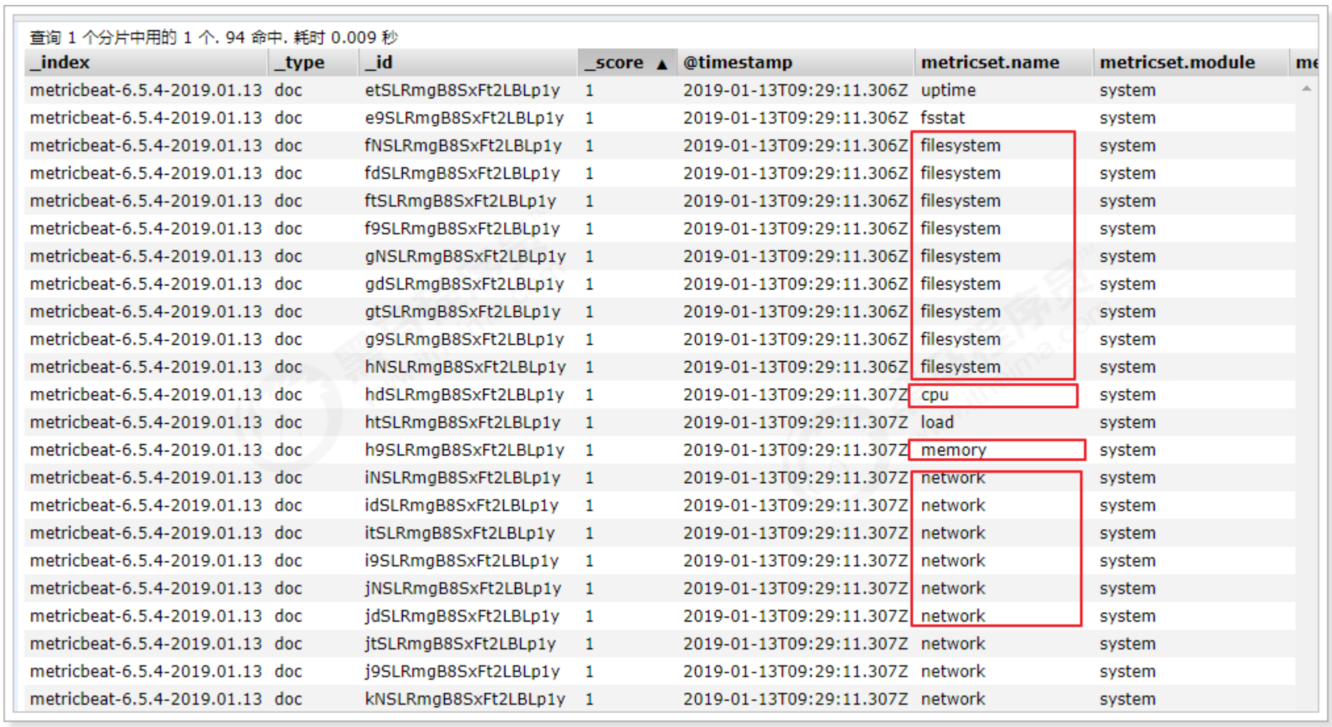
以Redis Module为例：



3.2、部署与收集系统指标

```
tar -xvf metricbeat-6.5.4-linux-x86_64.tar.gz
cd metricbeat-6.5.4-linux-x86_64
vim metricbeat.yml

metricbeat.config.modules:
  path: ${path.config}/modules.d/*.yml
  reload.enabled: false
setup.template.settings:
  index.number_of_shards: 1
  index.codec: best_compression
setup.kibana:
output.elasticsearch:
  hosts: ["192.168.1.7:9200","192.168.1.7:9201","192.168.1.7:9202"]
processors:
  - add_host_metadata: ~
  - add_cloud_metadata: ~

#启动
./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：

```
root@itcast01:modules.d# cat system.yml
# Module: system
# Docs: https://www.elastic.co/guide/en/beats/metricbeat/6.5/metricbeat-module-system.html

- module: system
  period: 10s
  metricsets:
    - cpu
    - load
    - memory
    - network
    - process
    - process_summary
    #- core
    #- diskio
    #- socket
  process.include_top_n:
    by_cpu: 5      # include top 5 processes by CPU
    by_memory: 5   # include top 5 processes by memory

- module: system
  period: 1m
  metricsets:
    - filesystem
    - fsstat
  processors:
  - drop_event.when.regexp:
      system.filesystem.mount_point: '^/(sys|cgroup|proc|dev|etc|host|lib)($|/)'

- module: system
  period: 15m
  metricsets:
    - uptime

#- module: system
#  period: 5m
#  metricsets:
#    - raid
#  raid.mount_point: '/'
```

3.3、Module

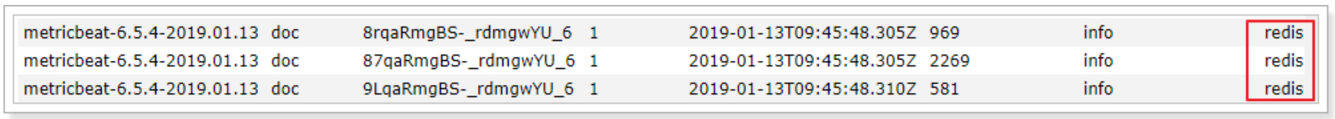
```
./metricbeat modules list  #查看列表

Enabled:
system #默认启用

Disabled:
aerospike
apache
ceph
couchbase
docker
dropwizard
elasticsearch
envoyproxy
etcd
golang
graphite
haproxy
http
jolokia
kafka
kibana
```

```
kubernetes
kvm
logstash
memcached
mongodb
munin
mysql
nginx
php_fpm
postgresql
prometheus
rabbitmq
redis
traefik
uwsgi
vsphere
windows
zookeeper
```

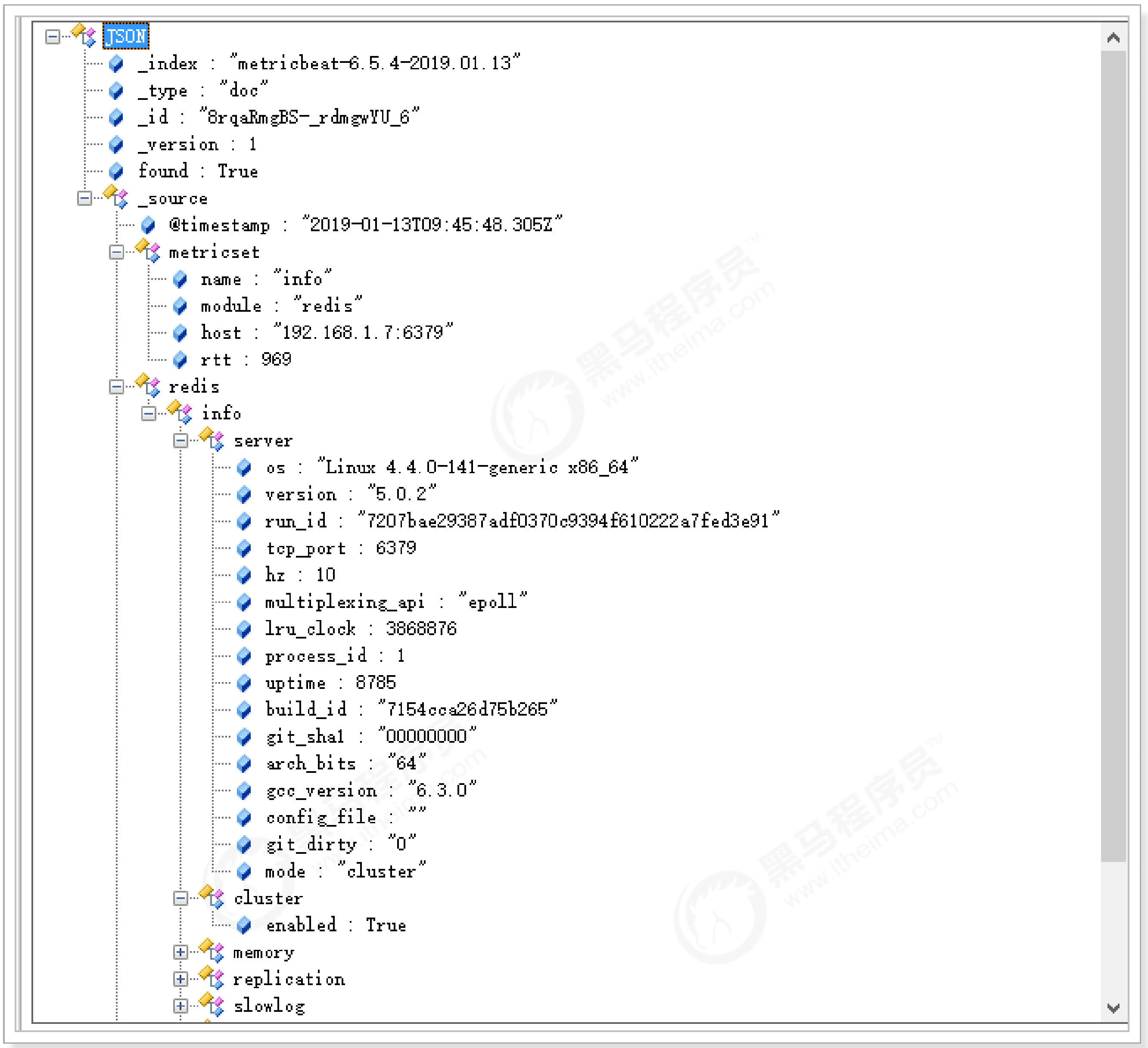
3.4、Redis Module

```
#启用redis module
./metricbeat modules enable redis
#修改redis module配置
vim modules.d/redis.yml

- module: redis
  metricsets:
    - info
  #  - keyspace
  period: 10s

  # Redis hosts
  hosts: ["192.168.1.7:6379","192.168.1.7:6380","192.168.1.7:6381"]

  # Network type to be used for redis connection. Default: tcp
  #network: tcp

  # Max number of concurrent connections. Default: 10
  #maxconn: 10

  # Redis AUTH password. Empty by default.
  #password: foobared

#启动
./metricbeat -e
```

可以看到Redis的指标数据写入到了Elasticsearch中：



```
metricbeat-6.5.4-2019.01.13  doc  8rqaRmgBS-_rdmgwYU_6  1  2019-01-13T09:45:48.305Z  969   info  redis
metricbeat-6.5.4-2019.01.13  doc  87qaRmgBS-_rdmgwYU_6  1  2019-01-13T09:45:48.305Z  2269  info  redis
metricbeat-6.5.4-2019.01.13  doc  9LqaRmgBS-_rdmgwYU_6  1  2019-01-13T09:45:48.310Z  581   info  redis
```



```
JSON
  _index : "metricbeat-6.5.4-2019.01.13"
  _type : "doc"
  _id : "8rqaRmgBS-_rdmgwYU_6"
  _version : 1
  found : True
  _source
    @timestamp : "2019-01-13T09:45:48.305Z"
    metricset
      name : "info"
      module : "redis"
      host : "192.168.1.7:6379"
      rtt : 969
    redis
      info
        server
          os : "Linux 4.4.0-141-generic x86_64"
          version : "5.0.2"
          run_id : "7207bae29387adf0370c9394f610222a7fed3e91"
          tcp_port : 6379
          hz : 10
          multiplexing_api : "epoll"
          lru_clock : 3868876
          process_id : 1
          uptime : 8785
          build_id : "7154cca26d75b265"
          git_sha1 : "00000000"
          arch_bits : "64"
          gcc_version : "6.3.0"
          config_file : ""
          git_dirty : "0"
          mode : "cluster"
        cluster
          enabled : True
        memory
        replication
        slowlog
```

更多的Module使用参见官方文档：

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

4、Kibana

您使用

Elastic Stack 的窗口

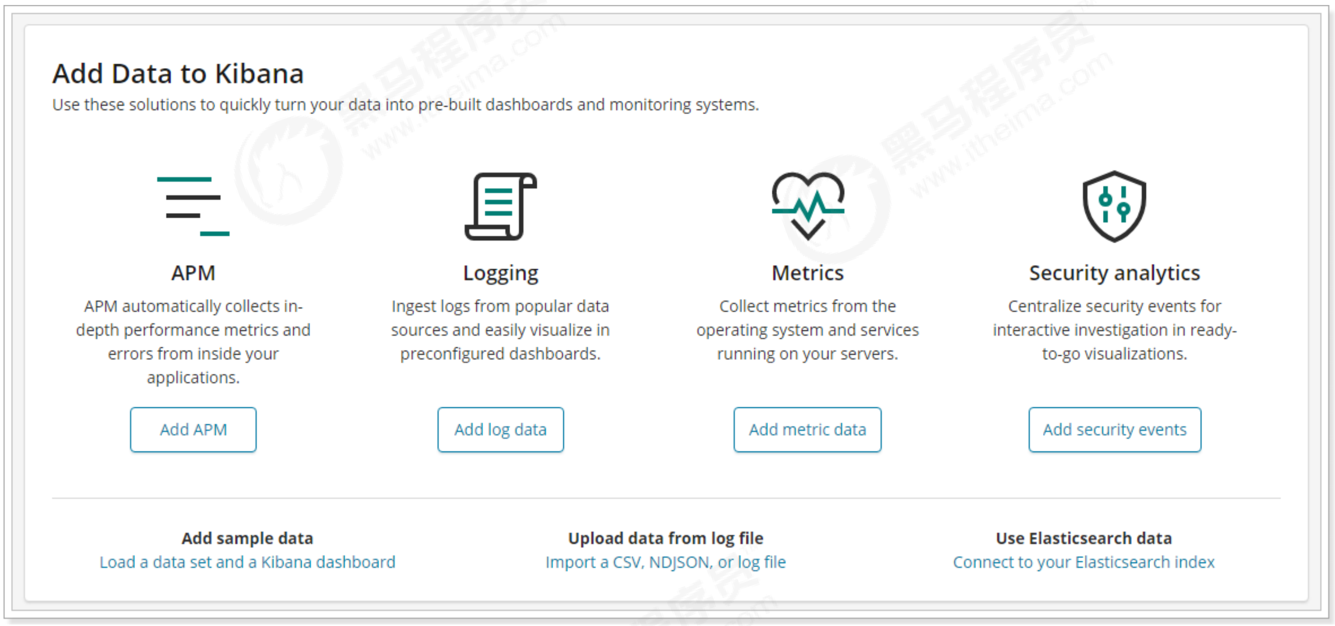
通过 Kibana，您能够对 Elasticsearch 中的数据进行可视化并在 Elastic Stack 进行操作，因此您可以在这里解开任何疑问：例如，为何会在凌晨 2:00 收到传呼，雨水会对季度数据造成怎样的影响。

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

4.1、配置安装

```
#解压安装包
tar -xvf kibana-6.5.4-linux-x86_64.tar.gz

#修改配置文件
vim config/kibana.yml

server.host: "192.168.1.7"  #对外暴露服务的地址
elasticsearch.url: "http://192.168.1.7:9200"  #配置Elasticsearch

#启动
./bin/kibana

#通过浏览器进行访问
http://192.168.1.7:5601/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

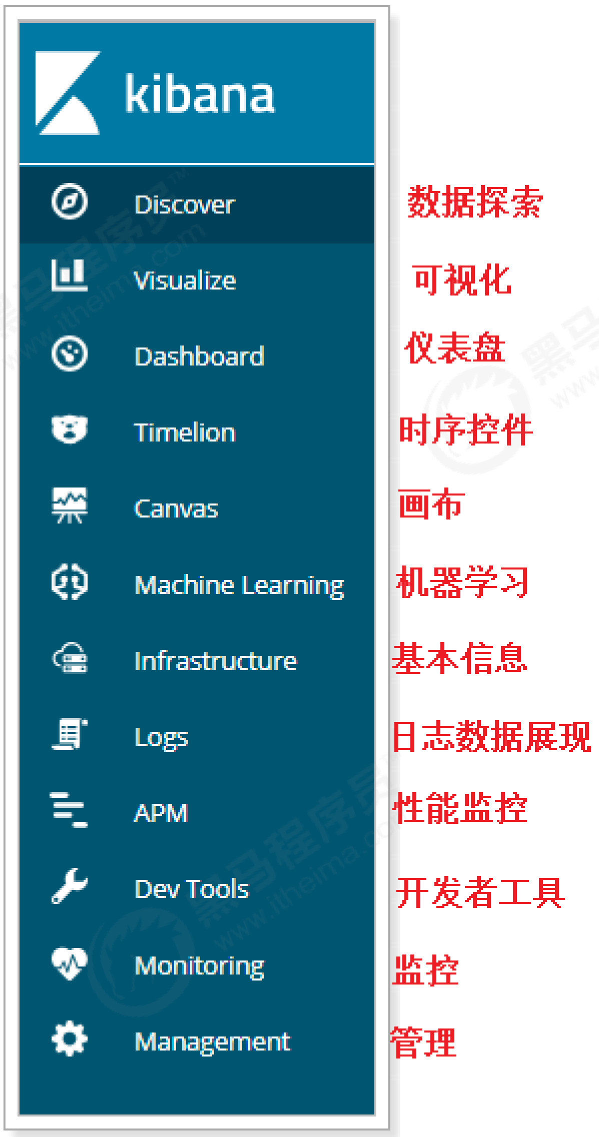
Connect to your Elasticsearch index

可以看到kibana页面，并且可以看到提示，导入数据到Kibana。

4.2、通过docker部署

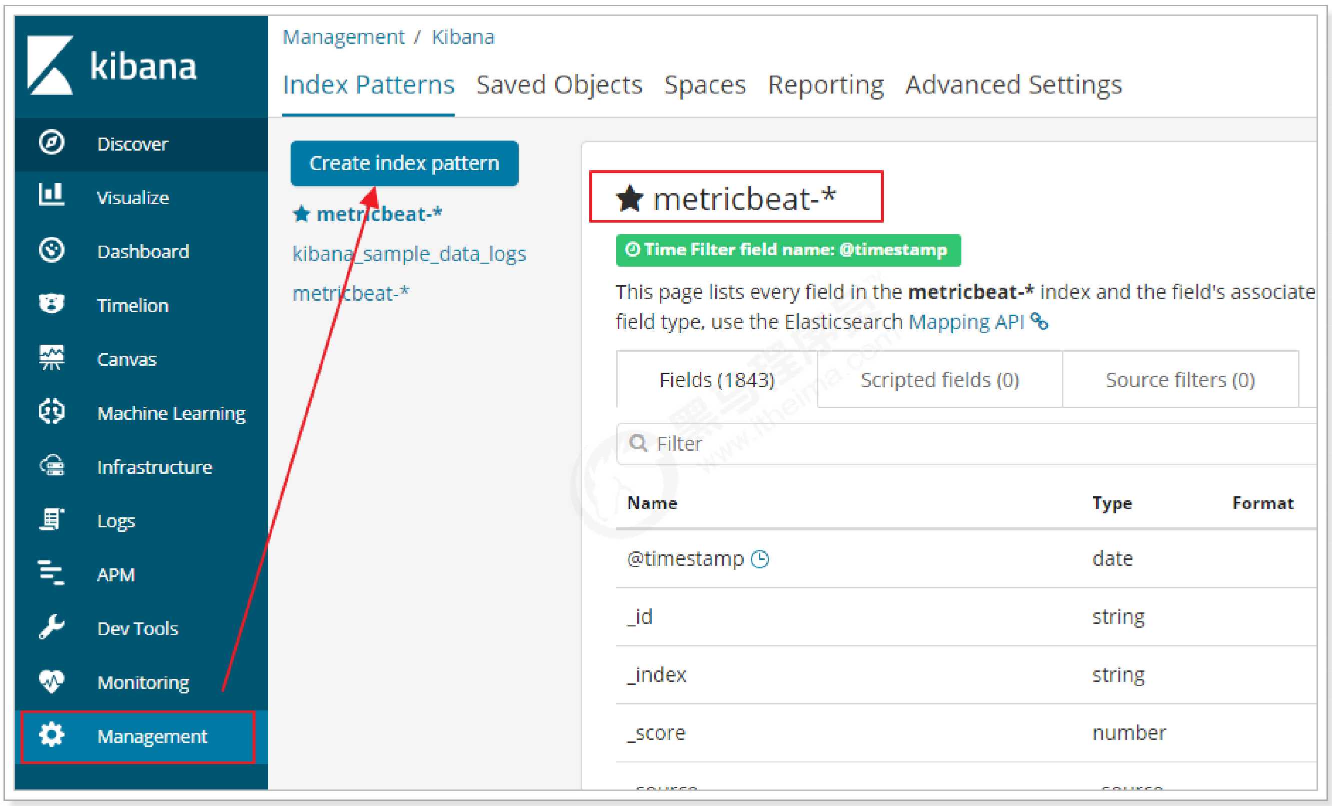
```
#拉取镜像
docker pull kibana:6.5.4

#创建配置文件
vim kibana.yml
server.host: "192.168.1.7"
elasticsearch.url: "http://192.168.1.7:9200"

#创建容器
docker create --name kibana --net host -v /haoke/beats/kibana-docker/kibana.yml:/usr/share/kibana/config/kibana.yml kibana:6.5.4

#启动容器
docker logs -f kibana
```

4.3、功能说明



4.4、数据探索

首先先添加索引信息：



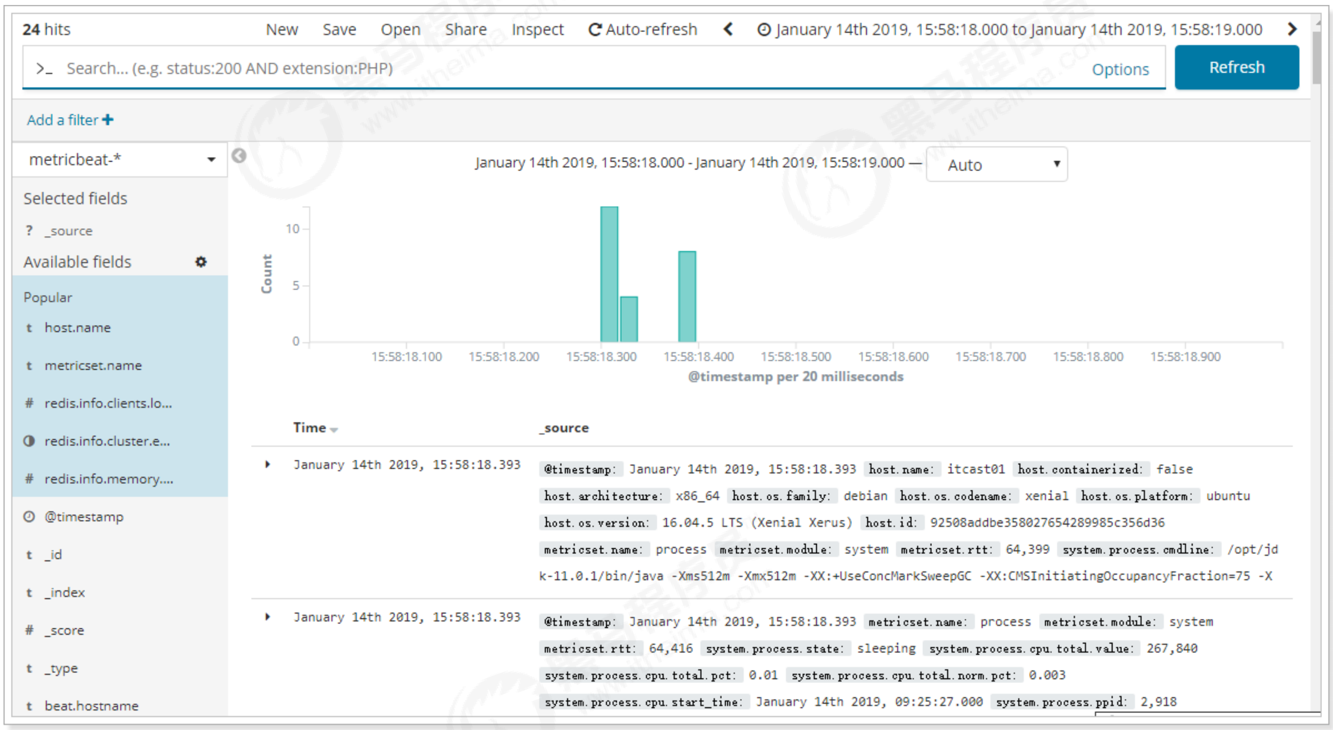
This page lists every field in the **metricbeat-*** index and the field's associate
field type, use the Elasticsearch Mapping API

Fields (1843) | Scripted fields (0) | Source filters (0)

Filter

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

即可查看索引数据：



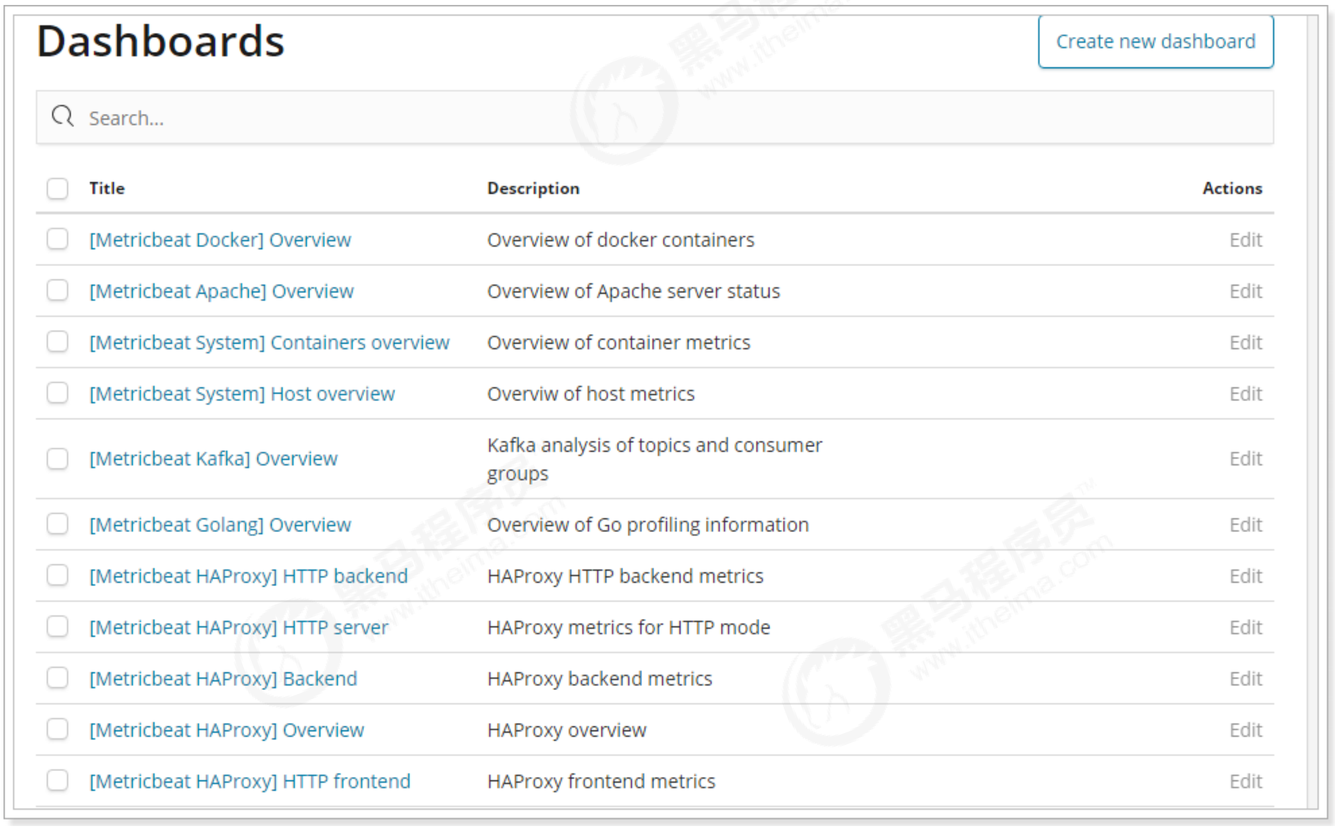
```
Time     _source
January 14th 2019, 15:58:18.393   @timestamp: January 14th 2019, 15:58:18.393 host.name: itcast01 host.containerized: false
                                  host.architecture: x86_64 host.os.family: debian host.os.codename: xenial host.os.platform: ubuntu
                                  host.os.version: 16.04.5 LTS (Xenial Xerus) host.id: 92508addbe358027654289985c356d36
                                  metricset.name: process metricset.module: system metricset.rtt: 64,399 system.process.cmdline: /opt/jd
                                  k-11.0.1/bin/java -Xms512m -Xmx512m -XX:+UseConcMarkSweepGC -XX:CMSInitiatingOccupancyFraction=75 -X
January 14th 2019, 15:58:18.393   @timestamp: January 14th 2019, 15:58:18.393 metricset.name: process metricset.module: system
                                  metricset.rtt: 64,416 system.process.state: sleeping system.process.cpu.total.value: 267,840
                                  system.process.cpu.total.pct: 0.01 system.process.cpu.total.norm.pct: 0.003
                                  system.process.cpu.start_time: January 14th 2019, 09:25:27.000 system.process.ppid: 2,918
```

4.5、Metricbeat 仪表盘

可以将Metricbeat的数据在Kibana中展示。

```
#修改metricbeat配置
setup.kibana:
  host: "192.168.1.7:5601"

#安装仪表盘到Kibana
./metricbeat setup --dashboards

```

即可在Kibana中看到仪表盘数据：

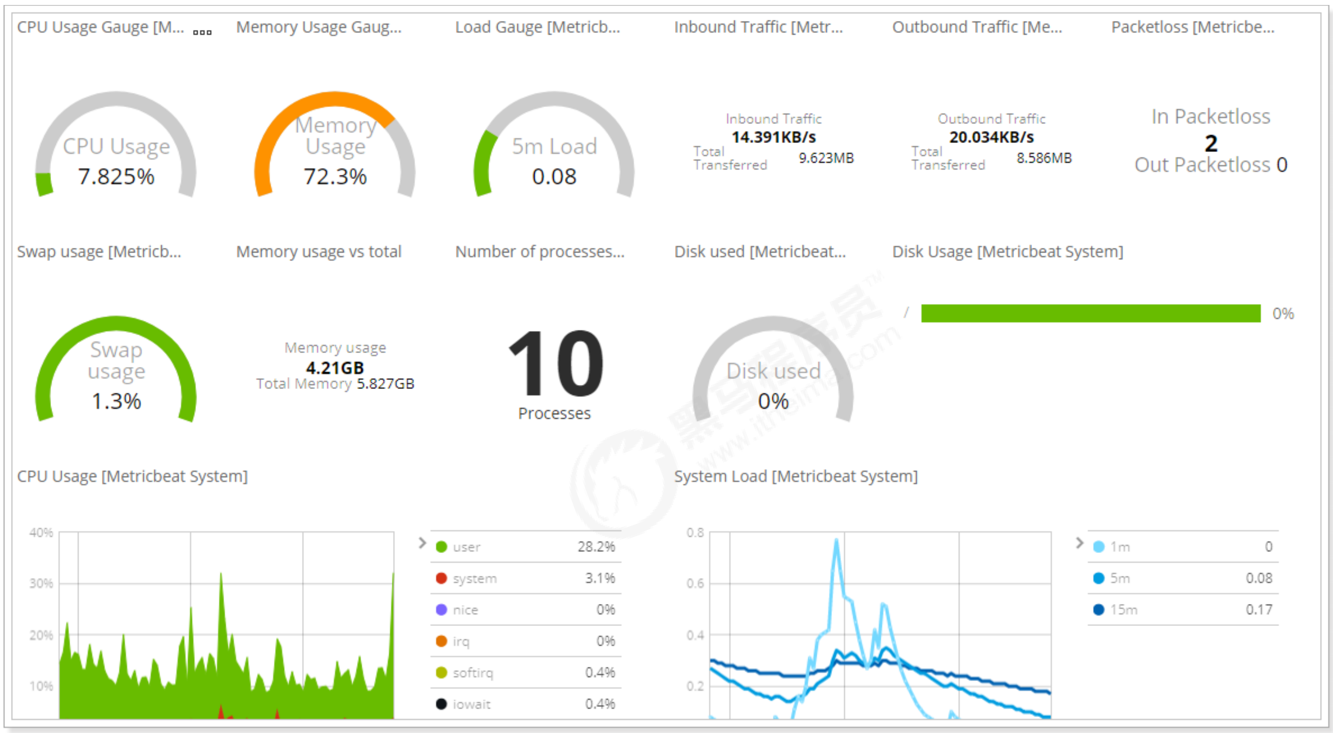

Dashboards

Create new dashboard

Search...

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

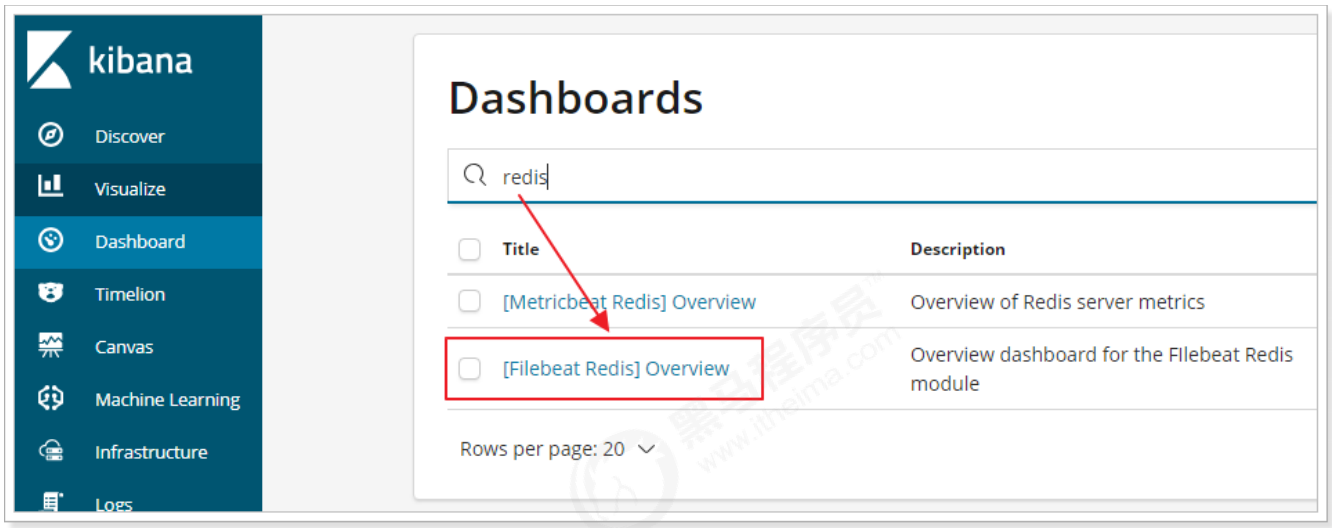
查看系统信息：



4.6、Filebeat 仪表盘

以Redis为例：

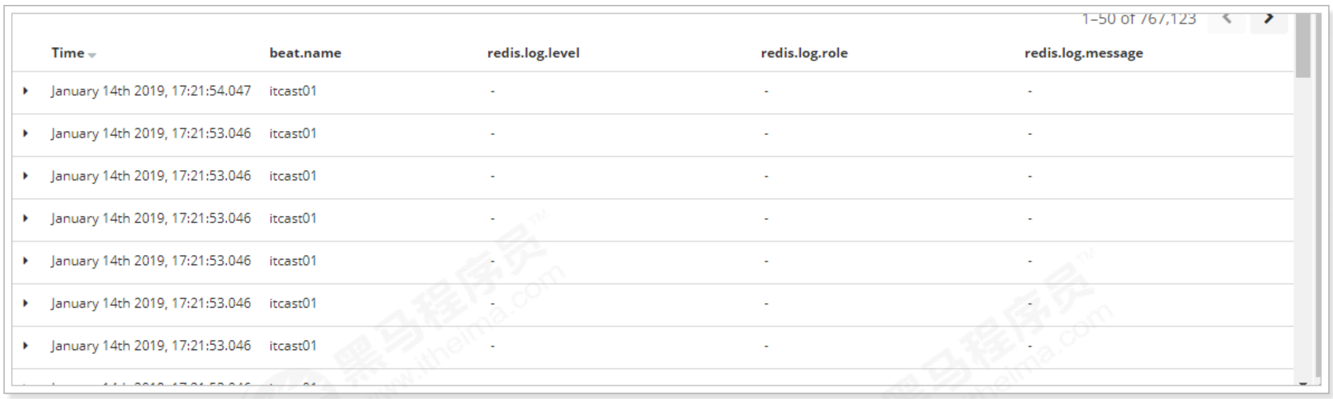
```
#修改配置文件
filebeat.inputs:
- type: log
  enabled: true
  paths:
     - /haoke/log/*.log
setup.template.settings:
  index.number_of_shards: 3
filebeat.config.modules:
  path: ${path.config}/modules.d/*.yml
  reload.enabled: false
output.elasticsearch:
  hosts: ["192.168.1.7:9200","192.168.1.7:9201","192.168.1.7:9202"]
setup.kibana:
  host: "192.168.1.7:5601"

#安装仪表盘到kibana
./filebeat -c haoke-redis.yml setup
```



Title	Description
[Metricbeat Redis] Overview	Overview of Redis server metrics
[Filebeat Redis] Overview	Overview dashboard for the Filebeat Redis module

Rows per page: 20

在kibana中已经看到了Filebeat中的redis仪表盘。



1–50 of 767,123

Time	beat.name	redis.log.level	redis.log.role	redis.log.message
January 14th 2019, 17:21:54.047	itcast01	-	-	-
January 14th 2019, 17:21:53.046	itcast01	-	-	-
January 14th 2019, 17:21:53.046	itcast01	-	-	-
January 14th 2019, 17:21:53.046	itcast01	-	-	-
January 14th 2019, 17:21:53.046	itcast01	-	-	-
January 14th 2019, 17:21:53.046	itcast01	-	-	-
January 14th 2019, 17:21:53.046	itcast01	-	-	-

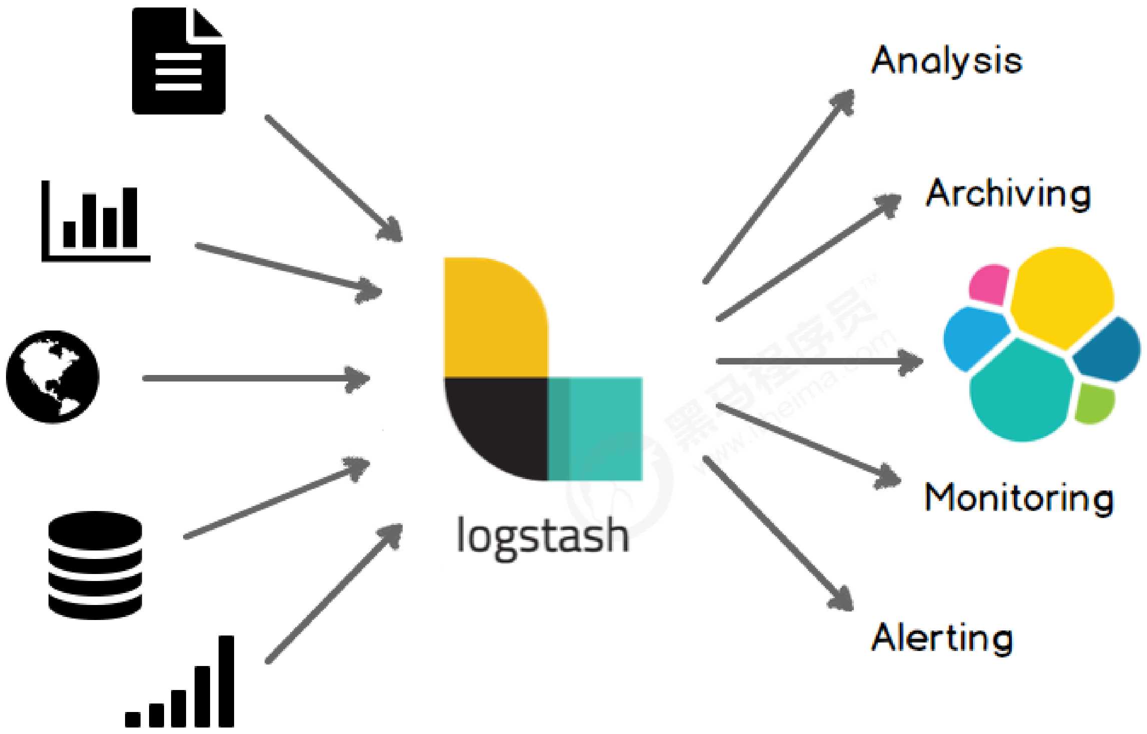
5、Logstash

5.1、简介

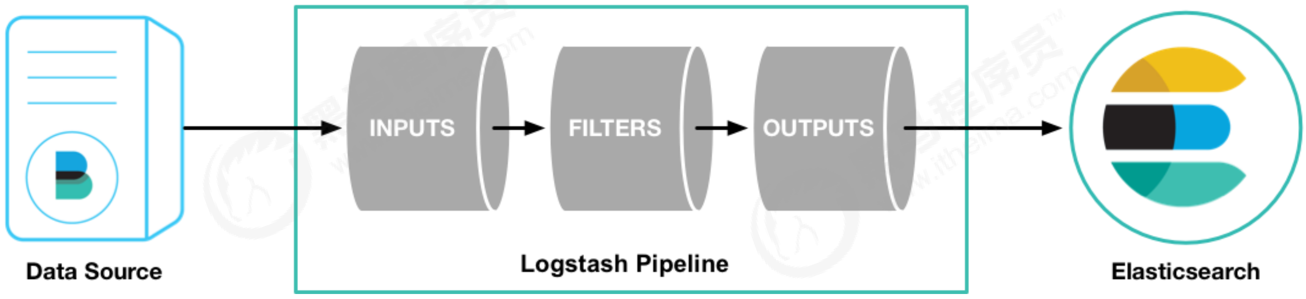
集中、转换和存储数据

Logstash 是开源的服务器端数据处理管道，能够同时从多个来源采集数据，转换数据，然后将数据发送到您最喜欢的“存储库”中。（我们的存储库当然是 Elasticsearch。）

用途：

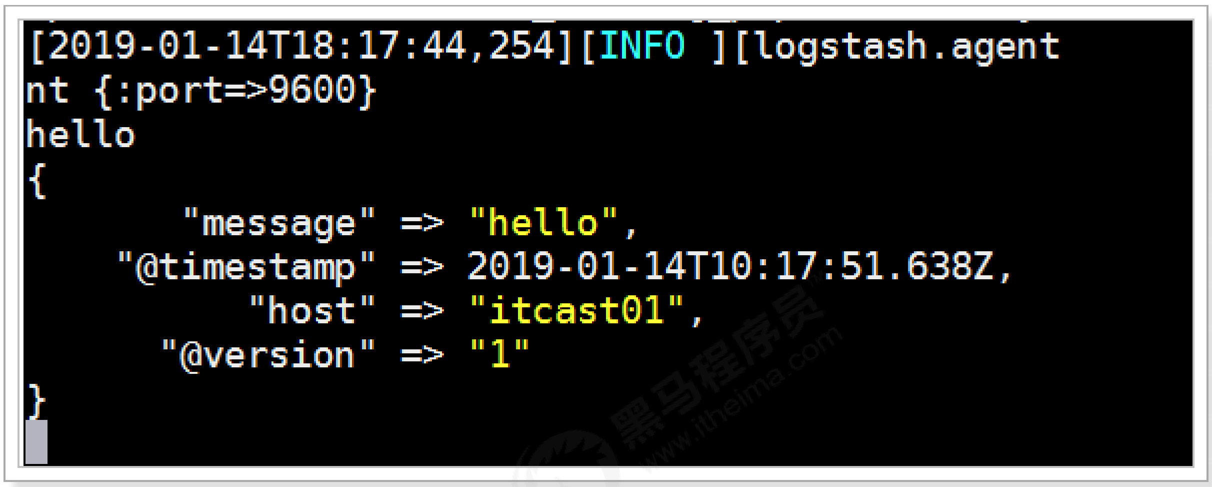


5.2、部署安装



```
#检查jdk环境，要求jdk1.8+
java -version

#解压安装包
tar -xvf logstash-6.5.4.tar.gz

#第一个logstash示例
bin/logstash -e 'input { stdin { } } output { stdout {} }'
```

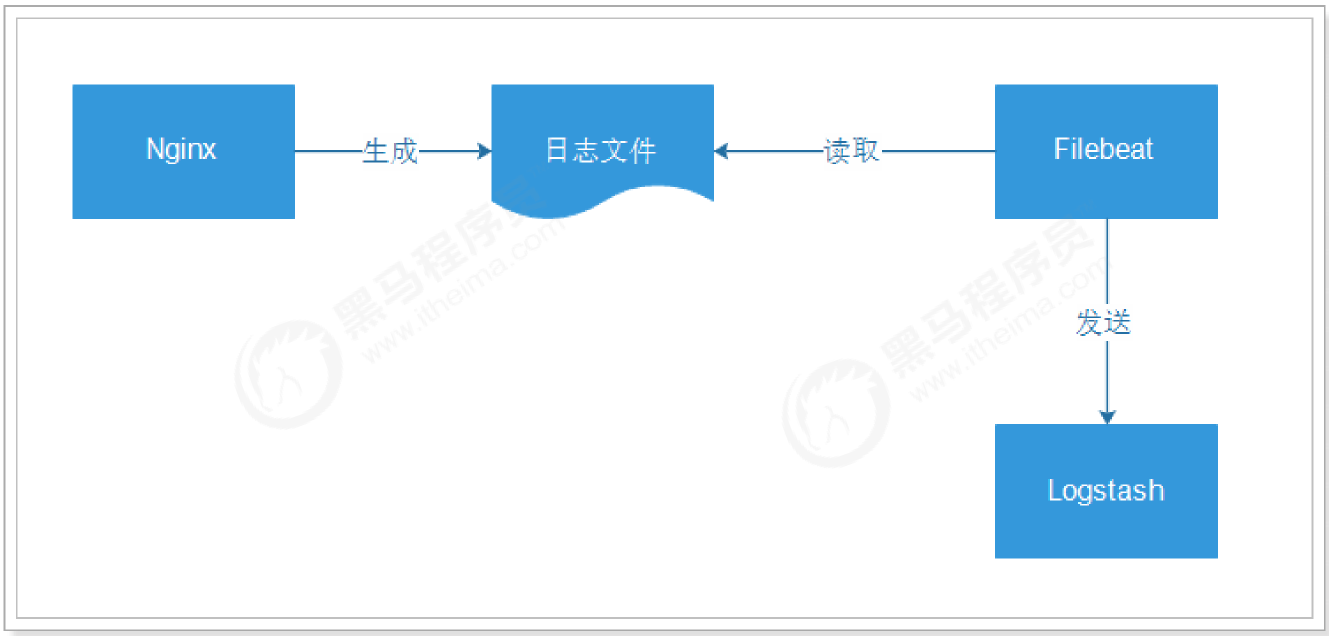
执行效果如下：



```
[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"
}
```

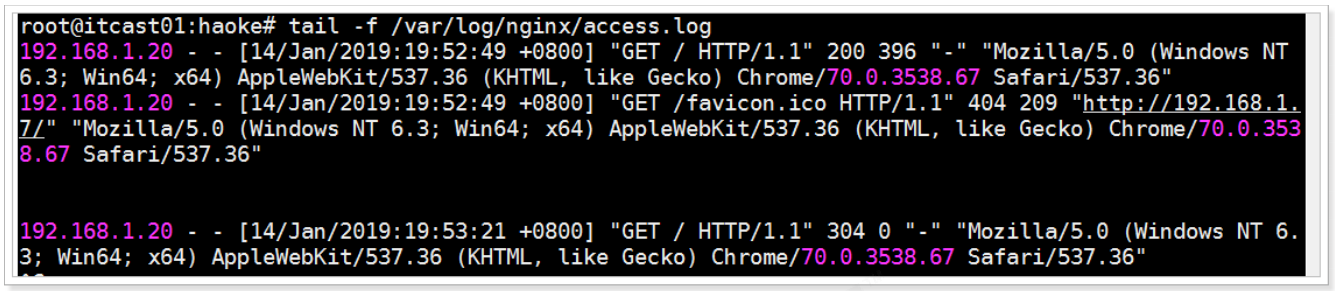
5.3、接收Filebeat输入的日志

接下来，我们将Filebeat和Logstash整合起来，读取nginx的日志。



5.3.1、安装Nginx

```
apt install nginx -y

#/usr/sbin/nginx：主程序
#/etc/nginx：存放配置文件
#/usr/share/nginx：存放静态文件
#/var/log/nginx：存放日志

#nginx服务命令
service nginx {start|stop|restart|reload|force-
reload|status|configtest|rotate|upgrade}

#通过浏览器访问页面并且查看日志
#访问地址：http://192.168.1.7/
tail -f /var/log/nginx/access.log
```



```
root@itcast01:haoke# tail -f /var/log/nginx/access.log
192.168.1.20 - - [14/Jan/2019:19:52:49 +0800] "GET / HTTP/1.1" 200 396 "-" "Mozilla/5.0 (Windows NT 6.3; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.67 Safari/537.36"
192.168.1.20 - - [14/Jan/2019:19:52:49 +0800] "GET /favicon.ico HTTP/1.1" 404 209 "http://192.168.1.7/" "Mozilla/5.0 (Windows NT 6.3; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.67 Safari/537.36"


192.168.1.20 - - [14/Jan/2019:19:53:21 +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"
```

5.3.2、配置Filebeat

```
#vim haoke-nginx.yml
filebeat.inputs:
- type: log
  enabled: true
  paths:
    - /var/log/nginx/access.log
  tags: ["log"]
  fields:
    from: nginx
  fields_under_root: false
output.logstash:
  hosts: ["192.168.1.7:5044"]

#启动
./filebeat -e -c haoke-nginx.yml
#说明：现在启动会报错，因为Logstash还没有启动
```

5.3.3、配置Logstash

```
vim haoke-pipeline.conf
#输入如下内容：
input {
    beats {
        port => "5044"
    }
}

# The filter part of this file is commented out to indicate that it is
# optional.
# filter {
#
# }

output {
    stdout { codec => rubydebug }
}

#启动 --config.test_and_exit 用于测试配置文件是否正确
bin/logstash -f haoke-pipeline.conf --config.test_and_exit

```

```
#[INFO ][logstash.runner          ] Using config.test_and_exit mode. Config
Validation Result: OK. Exiting Logstash

#正式启动 --config.reload.automatic 热加载配置文件，修改配置文件后无需重新启动
bin/logstash -f haoke-pipeline.conf --config.reload.automatic
```

5.3.4、测试

分别启动Filebeat和Logstash，刷新页面查看输出。

```
{
    "@timestamp" => 2019-01-14T12:23:37.604Z,
        "fields" => {
        "from" => "nginx"
    },
        "source" => "/var/log/nginx/access.log",
          "tags" => [
        [0] "log",
        [1] "beats_input_codec_plain_applied"
    ],
          "host" => {
        "name" => "itcast01"
    },
          "beat" => {
            "name" => "itcast01",
         "version" => "6.5.4",
        "hostname" => "itcast01"
    },
      "@version" => "1",
        "offset" => 600,
       "message" => "192.168.1.20 - - [14/Jan/2019:20:23:35 +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\"",
         "input" => {
        "type" => "log"
    },
    "prospector" => {
        "type" => "log"
    }
}

```

可以看到，已经在控制台输出了nginx的访问日志。

5.3.4、配置filter

在前面的输出中，可以看出，虽然可以拿到日志信息，但是信息格式并不友好，比如说，不能直接拿到日志中的ip地址。

> 第一步，自定义nginx的日志格式

```
vim /etc/nginx/nginx.conf

log_format main '$remote_addr - $remote_user [$time_local] '
                '"$request" $status $body_bytes_sent '
                '"$http_referer" "$http_user_agent"';

access_log /var/log/nginx/access.log main;

nginx -s reload
```

第二步，编写nginx-patterns文件

```
NGINX_ACCESS %{IPORHOST:remote_addr} - %{USERNAME:remote_user} \[%
{HTTPDATE:time_local}\] \"%{DATA:request}\" %{INT:status} %{NUMBER:bytes_sent} \"%
{DATA:http_referer}\" \"%{DATA:http_user_agent}\"
```

第三步，修改haoke-pipeline.conf文件

```
input {
    beats {
        port => "5044"
    }
}

filter {
    grok {
        patterns_dir => "/haoke/logstash-6.5.4/nginx-patterns"
        match => { "message" => "%{NGINX_ACCESS}"}
        remove_tag => [ "_grokparsefailure" ]
        add_tag => [ "nginx_access" ]
    }
}

output {
    stdout { codec => rubydebug }
}
```

第四步，测试

结果：

```
{
              "input" => {
        "type" => "log"
    },
       "http_referer" => "-",
        "remote_addr" => "192.168.1.20",
             "fields" => {
        "from" => "nginx"
    },
               "host" => {
```

```
         "name" => "itcast01"
    },
              "offset" => 664,
             "request" => "GET / HTTP/1.1",
             "message" => "192.168.1.20 - - [14/Jan/2019:22:52:49 +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\"",
          "bytes_sent" => "0",
              "status" => "304",
          "time_local" => "14/Jan/2019:22:52:49 +0800",
    "http_user_agent" => "Mozilla/5.0 (Windows NT 6.3; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.67 Safari/537.36",
          "prospector" => {
         "type" => "log"
    },
            "@version" => "1",
         "remote_user" => "-",
          "@timestamp" => 2019-01-14T14:52:58.948Z,
              "source" => "/var/log/nginx/access.log",
                "beat" => {
            "name" => "itcast01",
        "hostname" => "itcast01",
         "version" => "6.5.4"
    },
                "tags" => [
        [0] "log",
        [1] "beats_input_codec_plain_applied",
        [2] "nginx_access"
    ]
}

```

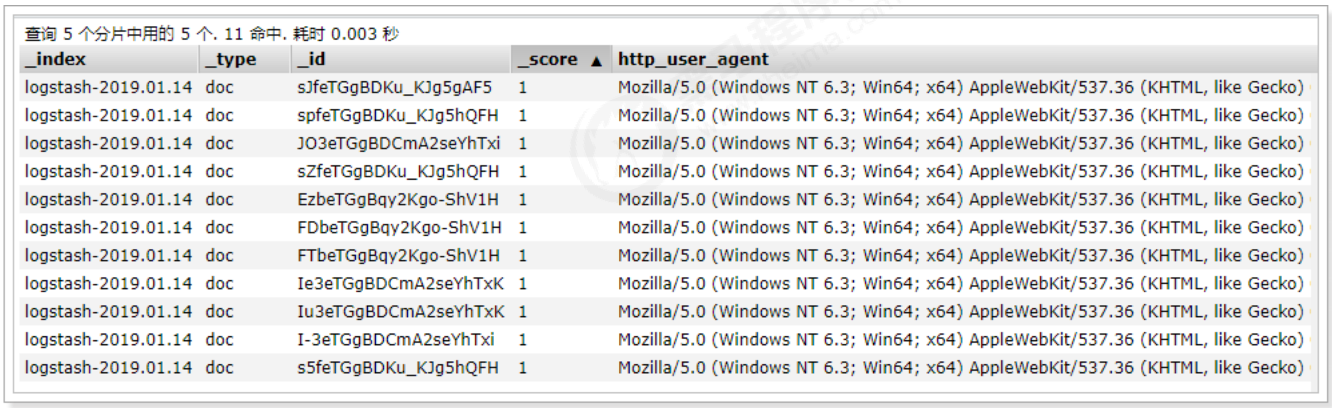
5.3.5、发送到Elasticsearch

修改配置：

```
#vim haoke-pipeline.conf
input {
    beats {
        port => "5044"
    }
}

filter {
    grok {
        patterns_dir => "/haoke/logstash-6.5.4/nginx-patterns"
        match => { "message" => "%{NGINX_ACCESS}"}
        remove_tag => [ "_grokparsefailure" ]
        add_tag => [ "nginx_access" ]
    }
}

#output {
```

```
#     stdout { codec => rubydebug }
#}
output {
    elasticsearch {
        hosts => [ "192.168.1.7:9200","192.168.1.7:9201","192.168.1.7:9202" ]
    }
}
```

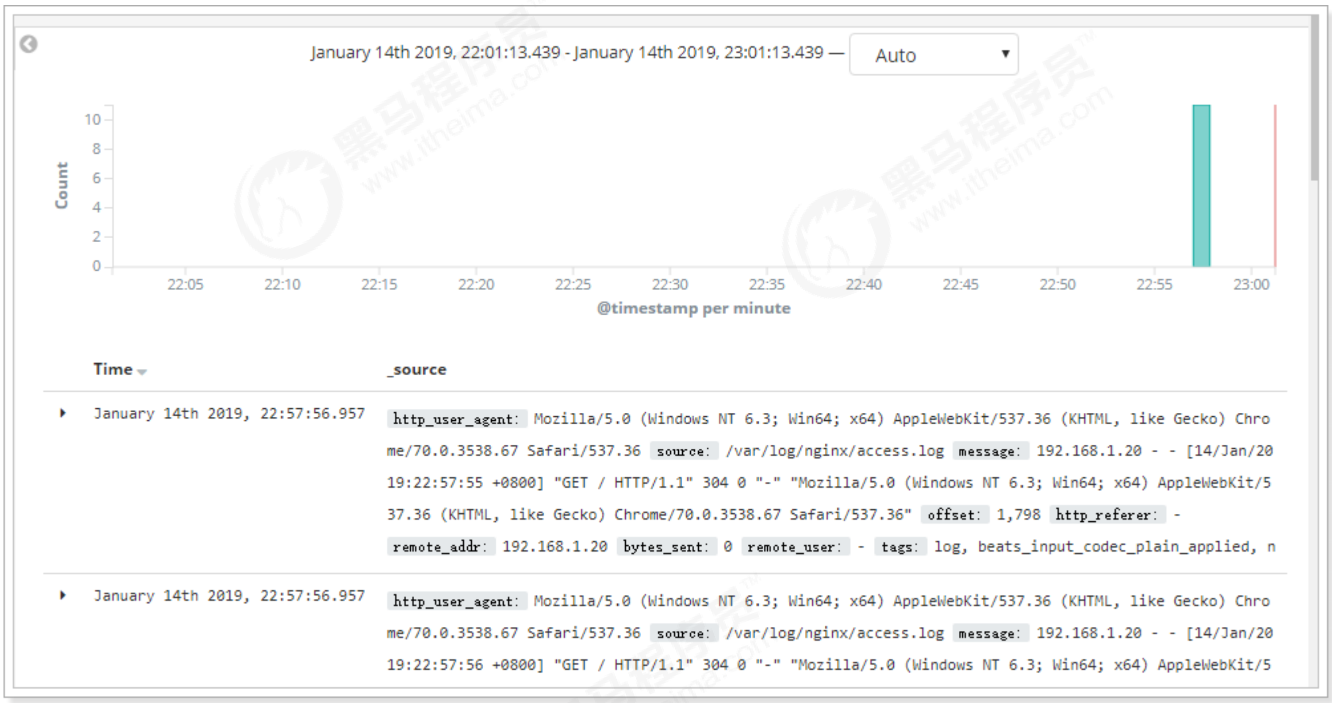
测试：



查询 5 个分片中用的 5 个. 11 命中. 耗时 0.003 秒

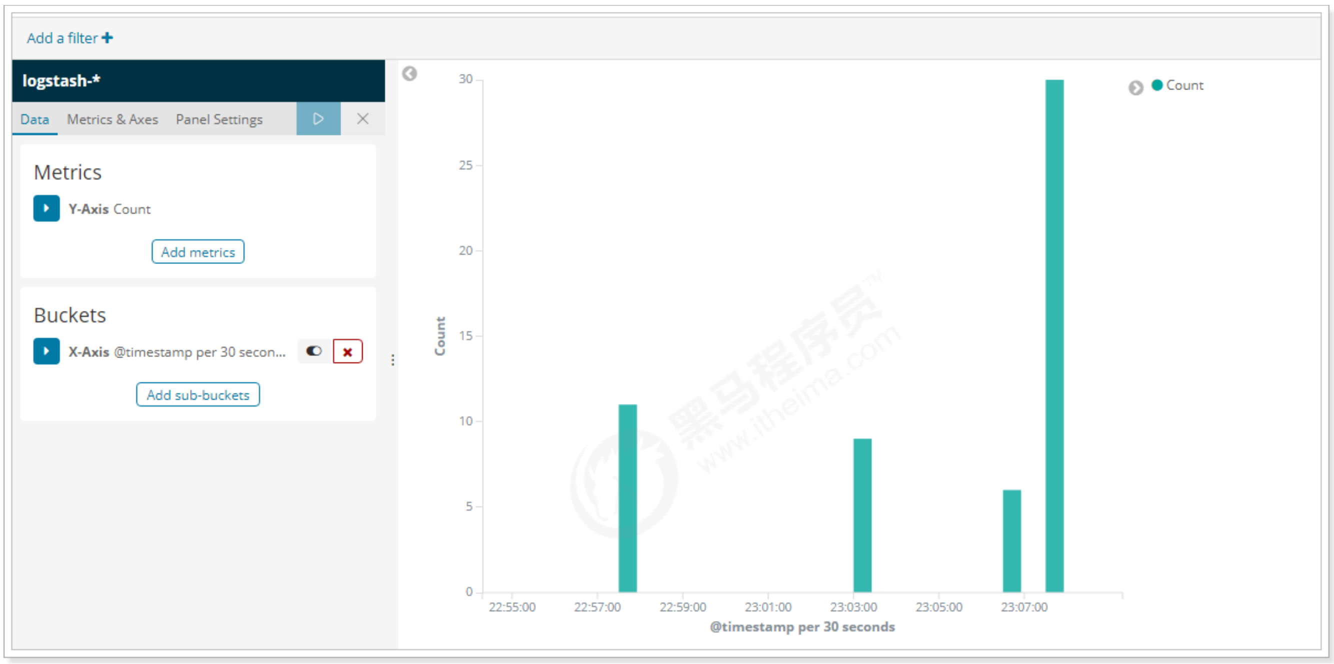
_index	_type	_id	_score ▲	http_user_agent
logstash-2019.01.14	doc	sJfeTGgBDKu_KJg5gAF5	1	Mozilla/5.0 (Windows NT 6.3; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
logstash-2019.01.14	doc	spfeTGgBDKu_KJg5hQFH	1	Mozilla/5.0 (Windows NT 6.3; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
logstash-2019.01.14	doc	JO3eTGgBDCmA2seYhTxi	1	Mozilla/5.0 (Windows NT 6.3; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
logstash-2019.01.14	doc	sZfeTGgBDKu_KJg5hQFH	1	Mozilla/5.0 (Windows NT 6.3; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
logstash-2019.01.14	doc	EzbeTGgBqy2Kgo-ShV1H	1	Mozilla/5.0 (Windows NT 6.3; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
logstash-2019.01.14	doc	FDbeTGgBqy2Kgo-ShV1H	1	Mozilla/5.0 (Windows NT 6.3; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
logstash-2019.01.14	doc	FTbeTGgBqy2Kgo-ShV1H	1	Mozilla/5.0 (Windows NT 6.3; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
logstash-2019.01.14	doc	Ie3eTGgBDCmA2seYhTxK	1	Mozilla/5.0 (Windows NT 6.3; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
logstash-2019.01.14	doc	Iu3eTGgBDCmA2seYhTxK	1	Mozilla/5.0 (Windows NT 6.3; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
logstash-2019.01.14	doc	I-3eTGgBDCmA2seYhTxi	1	Mozilla/5.0 (Windows NT 6.3; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
logstash-2019.01.14	doc	s5feTGgBDKu_KJg5hQFH	1	Mozilla/5.0 (Windows NT 6.3; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)

在Kibana中查看：



Time	_source
January 14th 2019, 22:57:56.957	http_user_agent: Mozilla/5.0 (Windows NT 6.3; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.67 Safari/537.36 source: /var/log/nginx/access.log message: 192.168.1.20 - - [14/Jan/2019:22:57:55 +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" offset: 1,798 http_referer: - remote_addr: 192.168.1.20 bytes_sent: 0 remote_user: - tags: log, beats_input_codec_plain_applied, n
January 14th 2019, 22:57:56.957	http_user_agent: Mozilla/5.0 (Windows NT 6.3; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.67 Safari/537.36 source: /var/log/nginx/access.log message: 192.168.1.20 - - [14/Jan/2019:22:57:56 +0800] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Windows NT 6.3; Win64; x64) AppleWebKit/5

制作柱形图：


Add a filter +

logstash-*

Data Metrics & Axes Panel Settings

Metrics
Y-Axis Count
Add metrics

Buckets
X-Axis @timestamp per 30 secon...
Add sub-buckets