# Elastic Stack

基础教程：<https://www.imooc.com/video/16131>

官方教程：<https://www.elastic.co/learn>

包括：Elasticsearch、Logstash、Beats、Kibana基础课程

<https://elasticsearch.cn/>

<https://elasticsearch.cn/explore/category-18>

# Elasticsearch And Kibana入门

## elasticsearch入门

官网：https://www.elastic.co/cn/

### 安装：

安装JDK1.8

http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html

下载安装Elasticsearch

https://www.elastic.co/downloads/elasticsearch

运行Elasticsearch

Run bin/elasticsearch (or bin\elasticsearch.bat on Windows)

### Elasticsearch配置说明：

配置文件位于config目录中：

elasticsearch.yml es的相关配置

jvm.options jvm的相关参数

log4j2.properties 日志相关配置

elasticsearch.yml关键配置说明：

cluster.name 集群名称，以此作为是否同一集群的判断条件

node.name 节点名称，以此作为集群中不同节点的区分条件

network.host/http.port 网络地址和端口，用于http和transport服务使用

path.data 数据储存地址

path.log 日志储存地址

elasticsearch有两种默认，Development与Production:

以transport的地址是否绑定在localhost为判断标准network.host

Development模式下在启动时会以warning的方式提示配置检测异常

Production默认下在启动时会以error的方式提示配置检查异常并退出

参数修改的第二方式：

在启动elasticsearch时配置：

bin/elasticsearch -Ehttp.port=19200

Elasticsearch本地启动集群的方式：

bin/elasticsearch

bin/elasticsearch -Ehttp.port=8200 -Epath.data=node2

bin/elasticsearch -Ehttp.port=7200 -Epath.data=node3

浏览器查看集群的配置信息：

127.0.0.1:8200/\_cat/nodes

127.0.0.1:8200/\_cat/nodes?v

127.0.0.1:8200/\_cluster/stats

## Kibana入门

### 安装

下载kibana

https://www.elastic.co/cn/downloads/kibana

运行：

1.Open config/kibana.yml in an editor

Set elasticsearch.url to point at your Elasticsearch instance

2.Run bin/kibana (or bin\kibana.bat on Windows)

3.Point your browser at http://localhost:5601

### Kibana配置说明：

配置位于config文件中

kibana.yml关键配置说明

server.host/server.port访问kibana用的地址和端口

elasticsearch.url待访问elasticsearch的地址

### Kibana常用功能说明

* Discover数据搜索查看
* Visualize图标制作
* Dashboard仪表盘制作
* Timelion时序数据的高级可视化分析
* DevTools开发者工具
* Management配置

## Elasticsearch术语及CURD操作

### 术语

Document文档数据；具体存在elasticsearch中的一条数据

Index索引；相当于mysql的数据库

Type索引中的数据类型；相当于mysql的table

Field字段，文档属性

Query DSL查询语法

### CURD操作

Elasticsearch CRUD

Create/Read/Update/Delete

Elasticsearch Create

|  |  |
| --- | --- |
| POST /accounts/person/1  {  "name":"John",  "lastname":"Doe",  "job\_description":"Systems administrator and Linux specialit"  } | {  "\_index": "accounts",  "\_type": "person",  "\_id": "1",  "\_version": 1,  "result": "created",  "\_shards": {  "total": 2,  "successful": 1,  "failed": 0  },  "\_seq\_no": 0,  "\_primary\_term": 1  } |

Elasticsearch Read

|  |  |
| --- | --- |
| GET /accounts/person/1 | {  "\_index": "accounts",  "\_type": "person",  "\_id": "1",  "\_version": 1,  "found": true,  "\_source": {  "name": "John",  "lastname": "Doe",  "job\_description": "Systems administrator and Linux specialit"  }  } |

Elasticsearch Update

|  |  |
| --- | --- |
| POST /accounts/person/1/\_update  {  "doc":{  "job\_description":"Systems administrator and Linux specialist"  }  } | {  "\_index": "accounts",  "\_type": "person",  "\_id": "1",  "\_version": 2,  "result": "updated",  "\_shards": {  "total": 2,  "successful": 1,  "failed": 0  },  "\_seq\_no": 1,  "\_primary\_term": 1  } |

Elasticsearch Delete

|  |  |
| --- | --- |
| DELETE /accounts/person/1  DELETE /accounts | {  "\_index": "accounts",  "\_type": "person",  "\_id": "1",  "\_version": 3,  "result": "deleted",  "\_shards": {  "total": 2,  "successful": 1,  "failed": 0  },  "\_seq\_no": 2,  "\_primary\_term": 1  } |

Elasticsearch Query

Query String:

GET /accounts/person/\_search?q="black"

Query DSL:

GET /accounts/person/\_search

{

"query": {

"match": {

"name": "black"

}

}

}

# Beats入门

## 简介

Lightweight Data Shipper

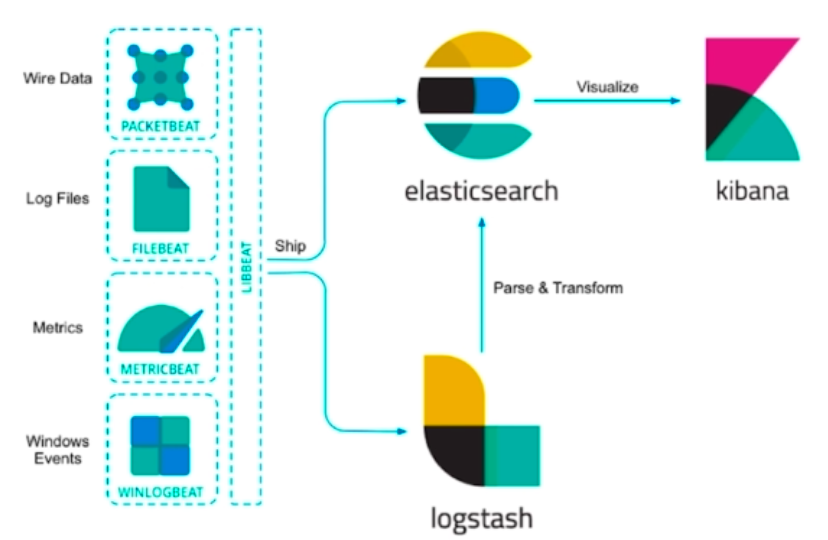
Filebeat 日志文件

Metricbeat 度量数据

Packetbeat 网络数据

Winlogbeat Windows数据

heartbeat 健康检查



## Filebeat入门

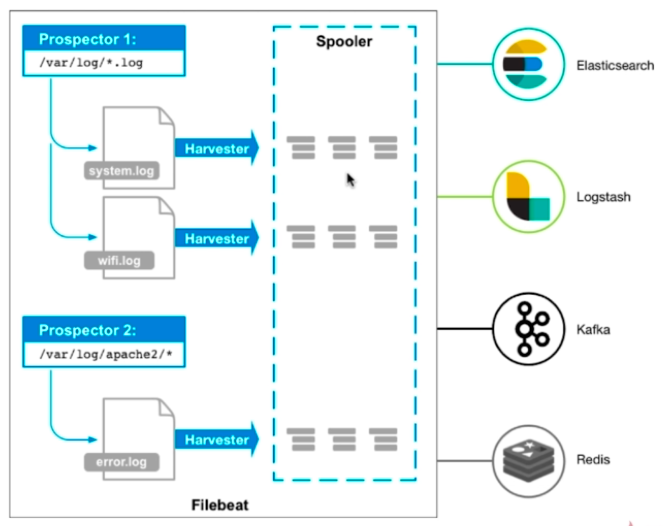
### Filebeat简介

Filebeat处理流程

Input 输入

Filter 处理

Output 输出



### Filebeat的配置

Filebeat Input配置

|  |  |
| --- | --- |
| yaml语法：  input\_type  log  stdin | filebeat.prospectors:  -input\_type:log  paths:  -/var/log/apache/httpd-\*.log  -input\_type:log  paths:  -/var/log/messages  -/var/log/\*.log |

Filebeat Output的配置

|  |  |
| --- | --- |
| Console  Elasticsearch  Logstash  Kafka  Redis  File | output.elasticsearch  hosts:[“http//localhost:9200”]  username:”admin”  password:”s3cr3t” |
| output.console:  pretty:true |

Filebeat Filter的配置

|  |  |
| --- | --- |
| Input时处理 | Output前处理--Processor |
| Include\_lines  exclude\_lines  exclude\_file | drop\_event  drop\_fields  Decode\_json\_fields  Include\_fields |

ex:

|  |  |
| --- | --- |
| processors:  -drop\_even  when:  regexp:  message:”^DBG:” | processors:  -decode\_json\_fields:  fields:[“inner”] |

{“outer”:”value”,”inner”:”{\”data\”:\”value\”}”} →

{“outer”:”value”,”inner”:”{”data”:”value”}”}

### Beat高级使用

Filebeat + Elasticsearch Ingest Node

Filebeat缺乏数据转换的能力

Elasticsearch Ingest Node

新增的node类型

在数据写入es前对数据进行处理转换

pipeline api

Filebeat Module

对于社区常见需求进行配置封装增加易用性：

nginx、apache、mysql

封装内容：

filebeat.yml配置 ingest node pipeline配置 Kibana dashboard

最佳实践参考：

### Demo

Filebeat收集nginx log

通过stdin收集日志

通过console输出结果

$ head -n 2 ~/Downloads/nginx\_logs/nginx.log //查看nginx日志

$ head -n 2 ~/Downloads/nginx\_logs/nginx.log | ./filebeat -e -c nginx.yml //使用filebeat

nginx.yem配置文件：

- input\_type: stdin

output.console:

pretty: true

## Packetbeat入门

### 简介

* 实时抓取网络包
* 自动解析应用层协议

ICMP(v4 and v6) / DNS / HTTP / Mysql / Redis / ...

* Wireshark

### Demo

***Packetbeat解析http协议***

1. 启动elasticsearch
2. 运行packetbeat

$ sudo ./packetbeat -e -c es.yml -strict.perms=false

1. es.yml配置文件：

packetbeat.interfaces.device: lo0

packetbeat.protocols.http:

ports: [9200]

send\_request: true

include\_body\_fro: [“application/json”,”x-www-form-urlencoded”]

==========================================================

output.console:

pretty:true

# Logstash入门

## 简介：

Data Shipper

ETL / Extract / Transform / Load / ...

Logstash is an open source, server-side data processing pipeline that ingests data from a multitude of sources simultaneously, transforms it, and then sends it to your favorite “stash.”

处理流程：

|  |  |  |
| --- | --- | --- |
| Input  file  redis  beats  kafka | Filter  grok  mutate  drop  date | Output  stdout  elasticsearch  redis  kafka |

Input/Output配置

input { file { path => “/tmp/abc.log” }}

output { stdout { codec => rubydebug }}

Filter配置

Grok

基于正则表达式提供了丰富可重用的模式（pattern）

基于此可以将非结构化数据作结构化处理

Date

将字符串类型的时间字段转换为时间戳类型，方便后续数据处理

Mutate

进行增加、修改、删除、替换等字段相关处理

......

Filter配置Grok示例：

55.3.244.1 GET /index.html 15824 0.043

↓

%{IP:client} %{WORD:method} %{URIPATHPARAM:request} %{NUMBER:bytes} %{NUMBER:duration}

%:说明是Grok → %{pattern:name} //按照pattern的格式解析数据，json的key为name

↓

{

“client” : “55.3.244.1” ,

“method” : “GET” ,

“request” : “/index.html” ,

“bytes” : 15842 ,

“duration” : 0.043

}

## Demo

收集nginx log

修改配置文件：

运行logstash

$ head -n 2 ~/Downloads/nginx\_logs/nginx.log | bin/logstash -f nginx\_logstash.conf

# 实战：分析Elasticsearch查询语句

目标：

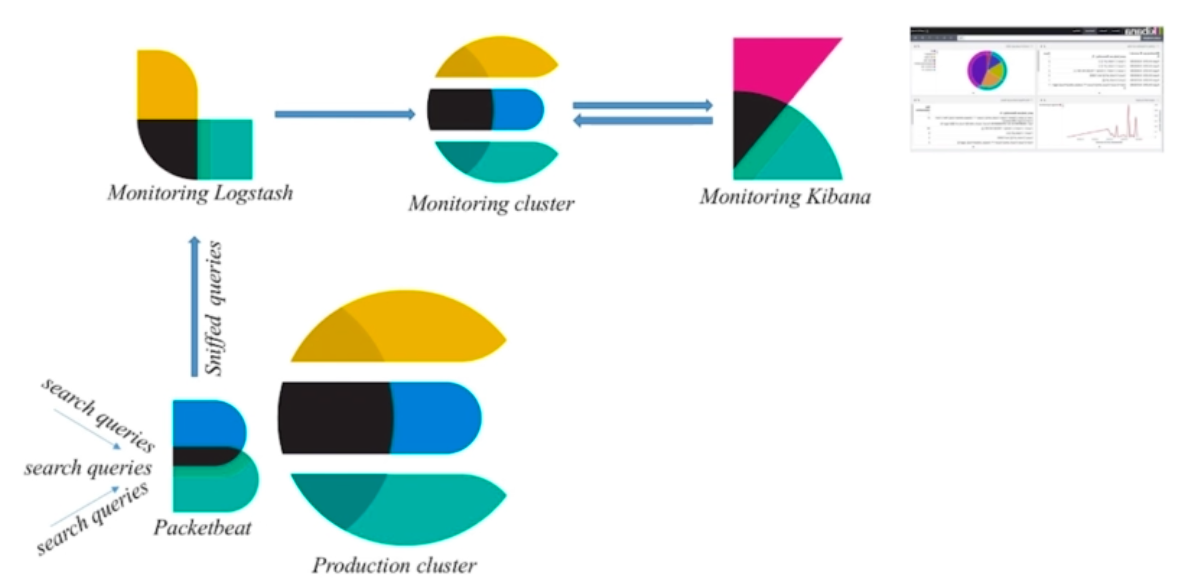
收集Elasticsearch集群的查询语句

分析查询语句的常用语句、响应时长等

方案：

应用Packetbeat + Logstash 完成数据收集工作

使用 kibana + Elasticsearch 完成数据分析工作



***方案：***

Production Cluster:

Elasticsearch <http://127.0.0.1:9200>

Kibana <http://127.0.0.1:5601>

Monitoring Cluster:

Elasticsearch <http://127.0.0.1:8200>

$ bin/elasticsearch -Ecluster.name=sniff\_search -Ehttp.port=8200 -Epath.data=sniff

Kibana <http://127.0.0.1:8601>

$ bin/kibana -e <http://127.0.0.1:8200> -p 8601

Production与Monitoring不能是一个集群，否则会进入抓包死循环

***Logstash***

input { beats { port => 5044 }}

filter {

if “search” in [request]{

grok { match => {“request” => “.\*\n\{(?<query\_body>.\*)”}}

grok { match => {“path” => “\/(?<index>.\*)\/\_search”}}

if [index] {} else { mutate { add\_field => {“index”=>”All”} }}

mutate {update => {“query\_body”=>”{%{query\_body}”}} }

}

output{

if “search” in [request]{

elasticsearch { hosts=>”127.0.0.1:8200”}

}}

***Packetbeat:***

packetbeat.interfaces.device:lo0

packetbeat.protocols.http: ports:[9200]

send\_request:true

include\_body\_for:[“application/json”,”x-www-form-urlendoded”]

output.logstash:hosts:[“127.0.0.1:5044”]

操作步骤：

启动生产集群：

$ bin/elasticsearch

$ bin/kibana

启动检测集群：

$ bin/elasticsearch -Ecluster.name=sniff\_search -Ehttp.port=8200 -Epath.data=sniff\_search

$ bin/kibana -e <http://127.0.0.1:8200> -p 8601

启动packetbeat：

$ sudo ./packetbeat -e -c sniff\_search.yml -strict.perms=false

启动logstash：

$ bin/logstash -f sniff\_search.conf