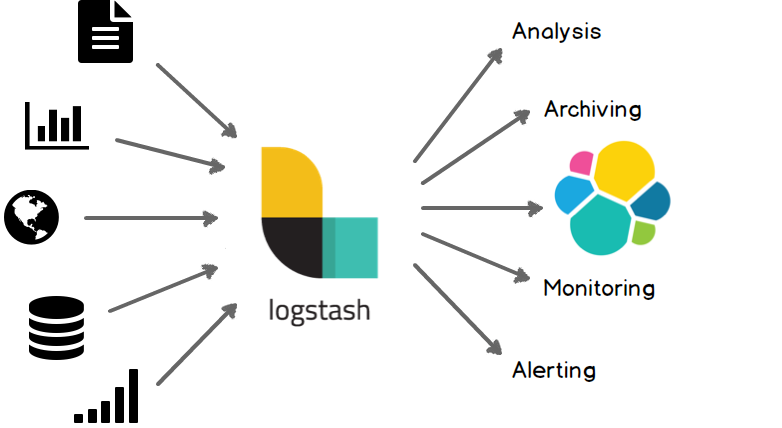
# **[Logstash](https://www.cnblogs.com/cjsblog/p/9459781.html)**

**Logstash介绍**

Logstash是一个开源数据收集引擎，具有实时管道功能。Logstash可以动态地将来自不同数据源的数据统一起来，并将数据标准化到你所选择的目的地。

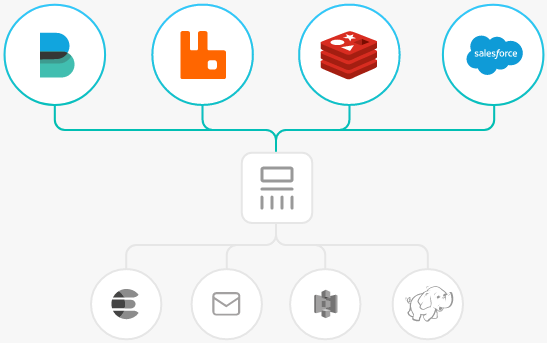


**集中、转换和存储你的数据**

 Logstash是一个开源的服务器端数据处理管道，可以同时从多个数据源获取数据，并对其进行转换，然后将其发送到你最喜欢的“存储”。（当然，我们最喜欢的是Elasticsearch）

**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。

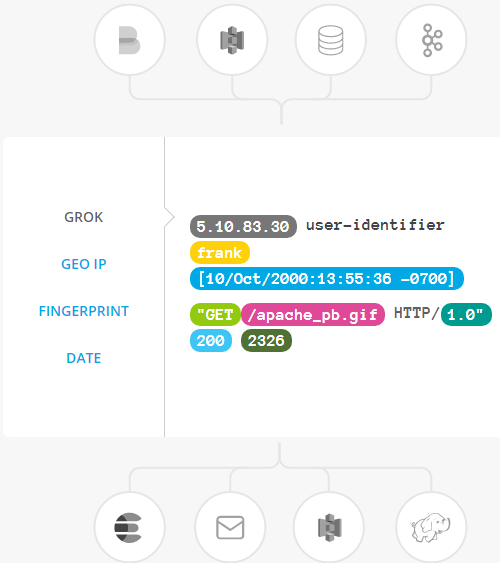


**过滤器：实时解析和转换数据**

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。

Logstash 能够动态地转换和解析数据，不受格式或复杂度的影响：

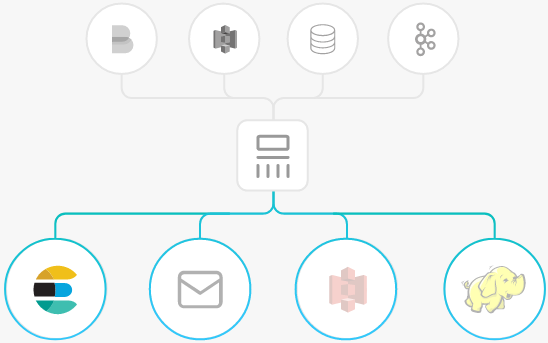
* 利用 Grok 从非结构化数据中派生出结构
* 从 IP 地址破译出地理坐标
* 将 PII 数据匿名化，完全排除敏感字段
* 整体处理不受数据源、格式或架构的影响



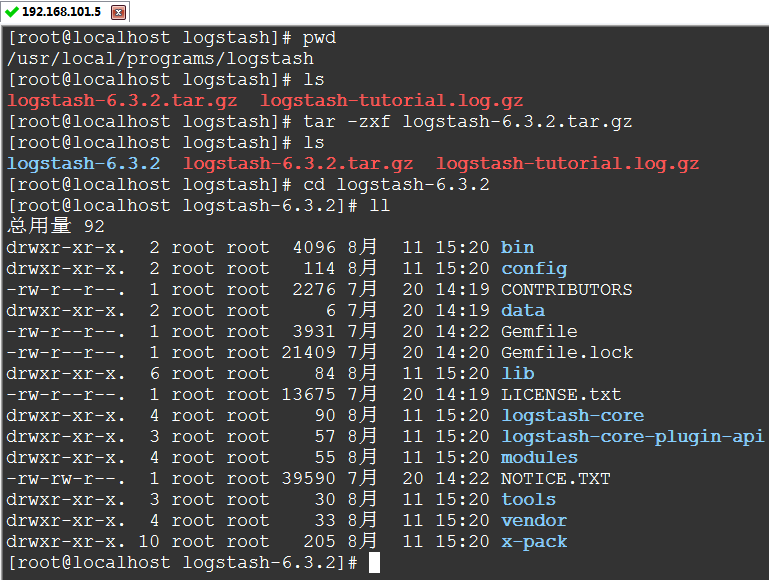
**输出：选择你的存储，导出你的数据**

尽管 Elasticsearch 是我们的首选输出方向，能够为我们的搜索和分析带来无限可能，但它并非唯一选择。

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。

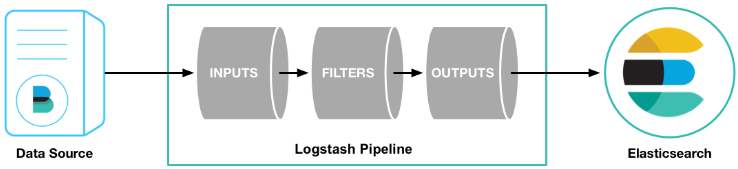


**安装Logstash**



首先，让我们通过最基本的Logstash管道来测试一下刚才安装的Logstash

Logstash管道有两个必需的元素，输入和输出，以及一个可选元素过滤器。输入插件从数据源那里消费数据，过滤器插件根据你的期望修改数据，输出插件将数据写入目的地。

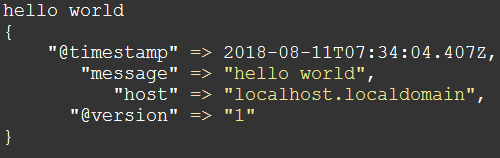


接下来，允许Logstash最基本的管道，例如：

bin/logstash -e 'input { stdin {} } output { stdout {} }'

（画外音：选项 -e 的意思是允许你从命令行指定配置）

启动以后，下面我们在命令行下输入"hello world"



**用Logstash解析日志**

 在上一小节中，你已经创建了一个基本的Logstash管道来测试你的Logstash设置。在现实世界中，一个Logstash管理会稍微复杂一些：它通常有一个或多个**input**, **filter** 和 **output** 插件。

在这一小节中，你将创建一个Logstash管道，并且使用Filebeat将Apache Web日志作为input，解析这些日志，然后将解析的数据写到一个Elasticsearch集群中。你将在配置文件中定义管道，而不是在命令行中定义管道配置。

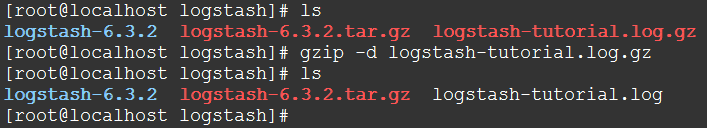
在开始之前，请先[下载示例数据](https://download.elastic.co/demos/logstash/gettingstarted/logstash-tutorial.log.gz" \t "https://www.cnblogs.com/cjsblog/p/_blank)。

**配置Filebeat来发送日志行到Logstash**

在你创建Logstash管道之前，你需要先配置Filebeat来发送日志行到Logstash。Filebeat客户端是一个轻量级的、资源友好的工具，它从服务器上的文件中收集日志，并将这些日志转发到你的Logstash实例以进行处理。Filebeat设计就是为了可靠性和低延迟。Filebeat在主机上占用的资源很少，而且Beats input插件将对Logstash实例的资源需求降到最低。

（画外音：注意，在一个典型的用例中，Filebeat和Logstash实例是分开的，它们分别运行在不同的机器上。在本教程中，Logstash和Filebeat在同一台机器上运行。）

关于Filebeat请参考《[开始使用Filebeat](https://www.cnblogs.com/cjsblog/p/9445792.html" \t "https://www.cnblogs.com/cjsblog/p/_blank)》



### **第1步：配置filebeat.yml**

[IMG_264](https://www.cnblogs.com/cjsblog/p/javascript:void(0);)

filebeat.inputs:

- type: log

paths:

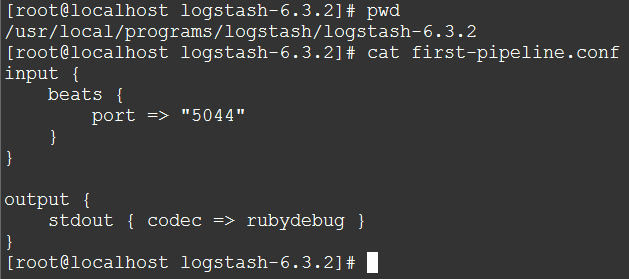
- /usr/local/programs/logstash/logstash-tutorial.log

output.logstash:

hosts: ["localhost:5044"]

[IMG_265](https://www.cnblogs.com/cjsblog/p/javascript:void(0);)

### **第2步：在logstash安装目录下新建一个文件first-pipeline.conf**



（画外音：刚才说过了通常Logstash管理有三部分（输入、过滤器、输出），这里input下面beats { port => "5044" }的意思是用Beats输入插件，而stdout { codec => rubydebug }的意思是输出到控制台）

### **第3步：检查配置并启动Logstash**

bin/logstash -f first-pipeline.conf --config.test\_and\_exit

（画外音：--config.test\_and\_exit选项的意思是解析配置文件并报告任何错误）

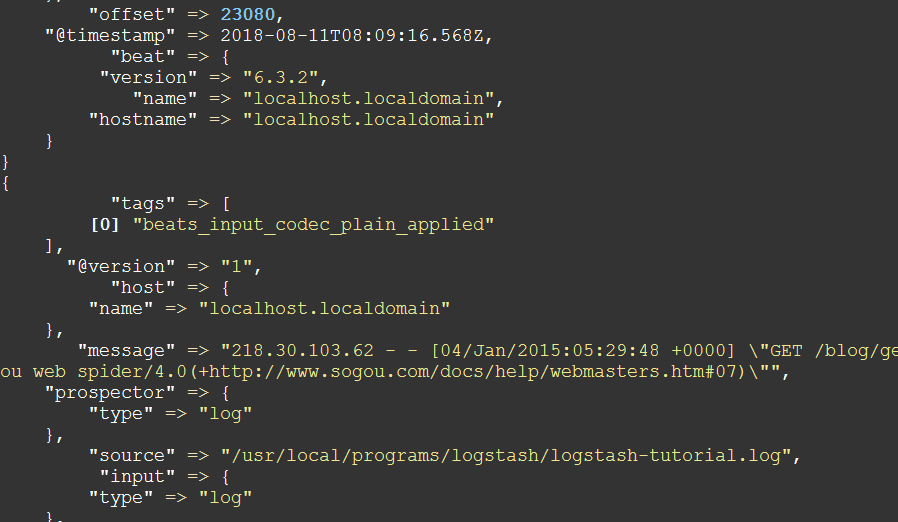
bin/logstash -f first-pipeline.conf --config.reload.automatic

（画外音：--config.reload.automatic选项的意思是启用自动配置加载，以至于每次你修改完配置文件以后无需停止然后重启Logstash）

### **第4步：启动filebeat**

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

如果一切正常，你将会在Logstash控制台下看到类似这样的输出：



**用Grok过滤器插件解析日志**

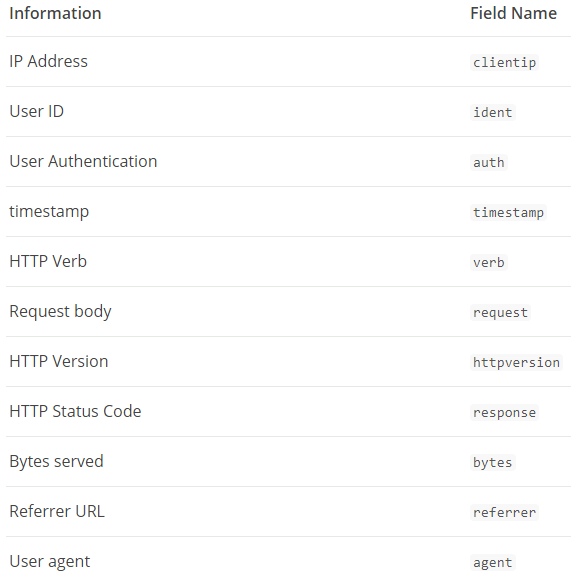
现在你有了一个工作管道，可以从Filebeat读取日志行。但是你可能已经注意到日志消息的格式并不理想。你想要解析日志消息，以便从日志中创建特定的、命名的字段。为此，您将使用grok filter插件。

grok 过滤器插件是Logstash中默认可用的几个插件之一。

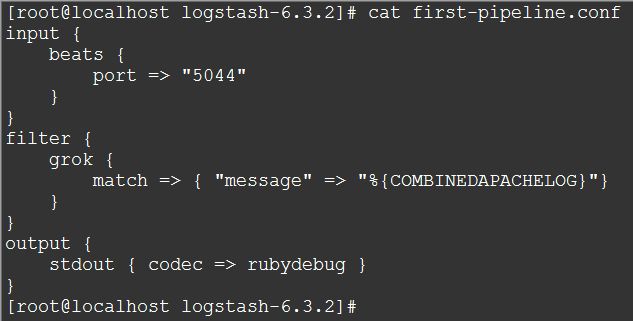
grok 过滤器插件允许你将非结构化日志数据解析为结构化和可查询的数据。

因为 grok 过滤器插件在传入的日志数据中查找模式

为了解析数据，你可以用 %{COMBINEDAPACHELOG} grok pattern ，这种模式（或者说格式）的schema如下：

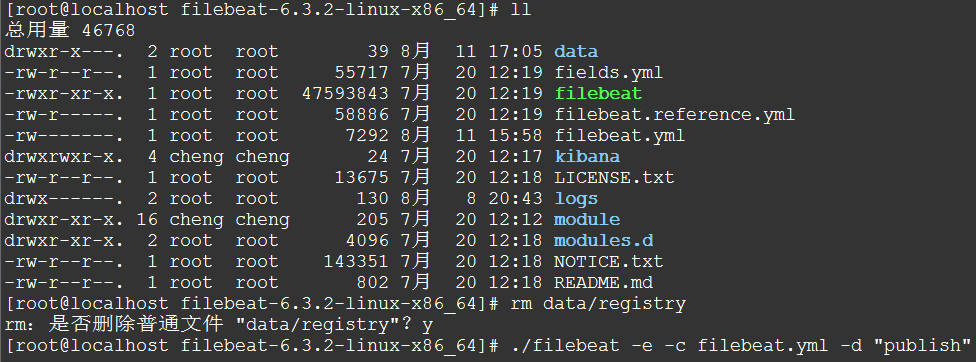


接下来，编辑first-pipeline.conf文件，加入grok filter，在你修改完以后这个文件看起来应该是这样的：



在你保存完以后，因为你已经启动了自动加载配置，所以你不需要重启Logstash来应用你的修改。但是，你确实需要强制Filebeat从头读取日志文件。为了这样做，你需要在终端先按下Ctrl+C停掉Filebeat，然后删除Filebeat注册文件。例如：

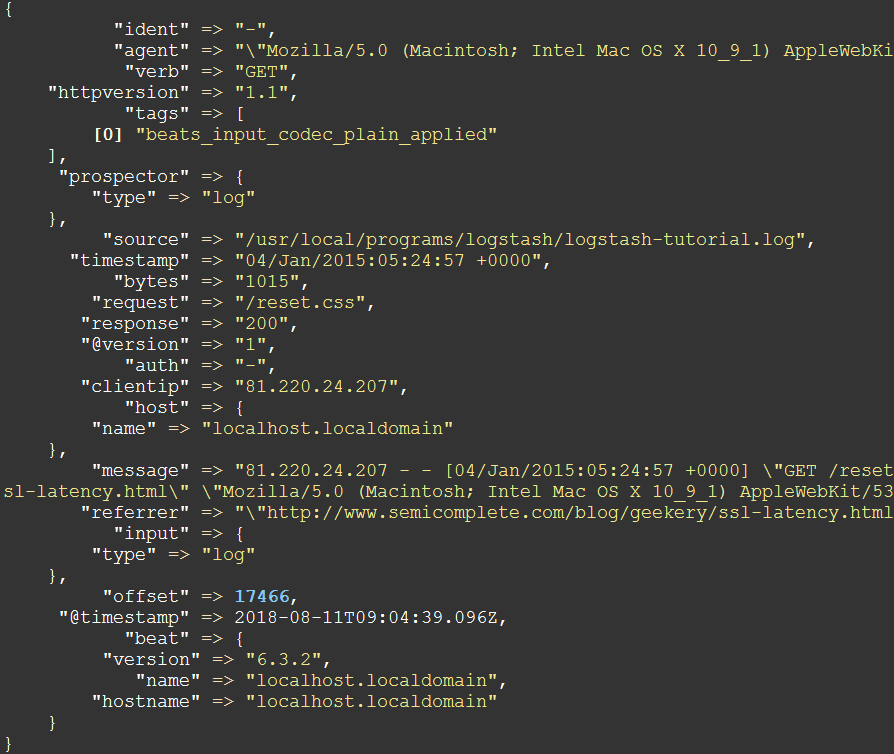
rm data/registr



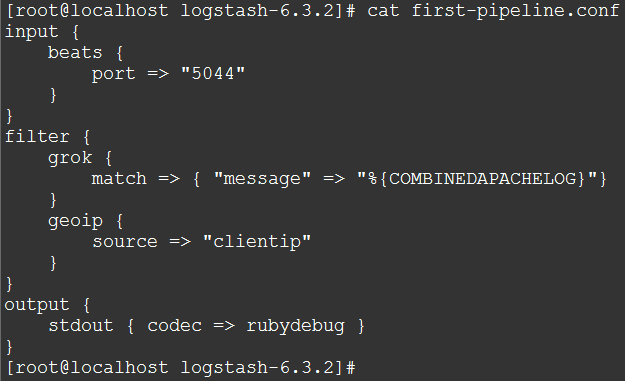
然后重启Filebeat

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

此时，再看Logstash控制台，输出可能是这样的：



**用 Geoip 过滤器插件增强你的数据**



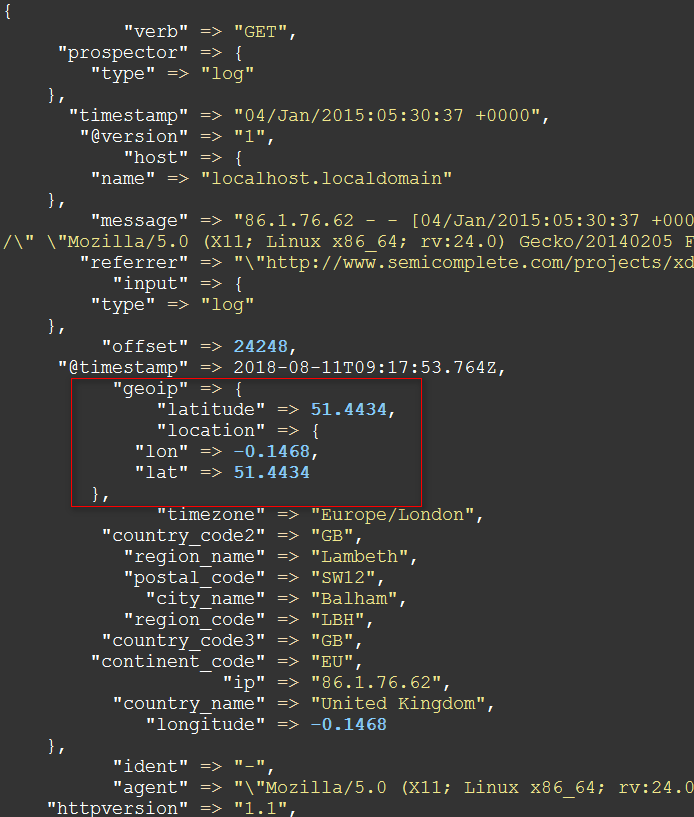
然后，同样地，重启Filebeat

Ctrl+C

rm data/registry

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

再次查看Logstash控制台，我们会发现多了地理位置信息：



**索引你的数据到Elasticsearch**

在之前的配置中，我们配置了Logstash输出到控制台，现在我们让它输出到Elasticsearch集群。

编辑first-pipeline.conf文件，替换output区域为：

output {

elasticsearch {

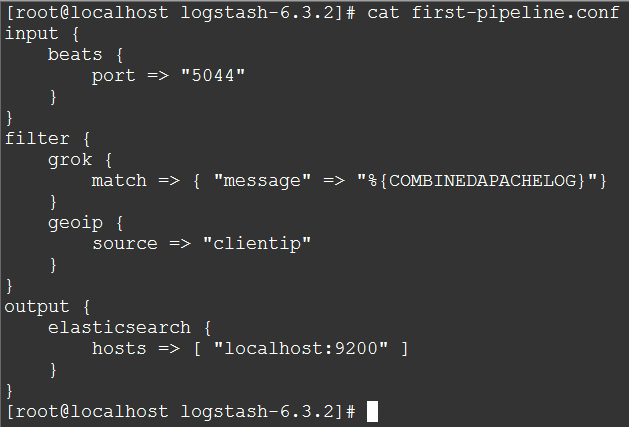
hosts => [ "localhost:9200" ]

}

}

在这段配置中，Logstash用http协议连接到Elasticsearch，而且假设Logstash和Elasticsearch允许在同一台机器上。你也可以指定一个远程的Elasticsearch实例，比如host=>["es-machine:9092"]

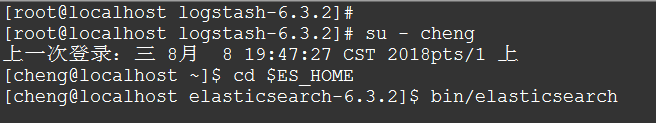
现在，first-pipeline.conf文件是这样的：



同样，保存改变以后，重启Filebeat

（画外音：首先，Ctrl+C终止Filebeat；接着rm data/registry删除注册文件；最后，./filebeat -e -c filebeat.yml -d "publish" 启动Filebeat）

好了，接下来启动Elasticsearch



（画外音：查看Elasticsearch索引，如果没有看到logstash的索引，那么重启Filebeat和Logstash，重启之后应该就可以看到了）

如果一切正常的话，可以在Elasticsearch的控制台日志中看到这样的输出：

[2018-08-11T17:35:27,871][INFO ][o.e.c.m.MetaDataIndexTemplateService] [Px524Ts] adding template [logstash] for index patterns [logstash-\*]

[2018-08-11T17:46:13,311][INFO ][o.e.c.m.MetaDataCreateIndexService] [Px524Ts] [logstash-2018.08.11] creating index, cause [auto(bulk api)], templates [logstash], shards [5]/[1], mappings [\_default\_]

[2018-08-11T17:46:13,549][INFO ][o.e.c.m.MetaDataMappingService] [Px524Ts] [logstash-2018.08.11/pzcVdNxSSjGzaaM9Ib\_G\_w] create\_mapping [doc]

[2018-08-11T17:46:13,722][INFO ][o.e.c.m.MetaDataMappingService] [Px524Ts] [logstash-2018.08.11/pzcVdNxSSjGzaaM9Ib\_G\_w] update\_mapping [doc]

这个时候，我们再查看Elasticsearch的索引

请求：

curl 'localhost:9200/\_cat/indices?v'

响应：

health status index uuid pri rep docs.count docs.deleted store.size pri.store.size

yellow open bank 59jD3B4FR8iifWWjrdMzUg 5 1 1000 0 475.1kb 475.1kb

green open .kibana DzGTSDo9SHSHcNH6rxYHHA 1 0 153 23 216.8kb 216.8kb

yellow open filebeat-6.3.2-2018.08.08 otgYPvsgR3Ot-2GDcw\_Upg 3 1 255 0 63.7kb 63.7kb

yellow open customer DoM-O7QmRk-6f3Iuls7X6Q 5 1 1 0 4.5kb 4.5kb

yellow open logstash-2018.08.11 pzcVdNxSSjGzaaM9Ib\_G\_w 5 1 100 0 251.8kb 251.8kb

可以看到有一个名字叫"logstash-2018.08.11"的索引，其它的索引都是之前建的不用管

接下来，查看这个索引下的文档

请求：

curl -X GET 'localhost:9200/logstash-2018.08.11/\_search?pretty&q=response=200'

响应大概是这样的：

（画外音：由于输出太长了，这里截取部分）

[IMG_276](https://www.cnblogs.com/cjsblog/p/javascript:void(0);)

{

"\_index" : "logstash-2018.08.11",

"\_type" : "doc",

"\_id" : "D\_JhKGUBOuOlYJNtDfwl",

"\_score" : 0.070617564,

"\_source" : {

"host" : {

"name" : "localhost.localdomain"

},

"httpversion" : "1.1",

"ident" : "-",

"message" : "83.149.9.216 - - [04/Jan/2015:05:13:42 +0000] \"GET /presentations/logstash-monitorama-2013/images/kibana-search.png HTTP/1.1\" 200 203023 \"http://semicomplete.com/presentations/logstash-monitorama-2013/\" \"Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_9\_1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/32.0.1700.77 Safari/537.36\"",

"auth" : "-",

"timestamp" : "04/Jan/2015:05:13:42 +0000",

"input" : {

"type" : "log"

},

"geoip" : {

"postal\_code" : "101194",

"region\_name" : "Moscow",

"timezone" : "Europe/Moscow",

"continent\_code" : "EU",

"city\_name" : "Moscow",

"country\_code3" : "RU",

"country\_name" : "Russia",

"ip" : "83.149.9.216",

"country\_code2" : "RU",

"region\_code" : "MOW",

"latitude" : 55.7485,

"longitude" : 37.6184,

"location" : {

"lon" : 37.6184,

"lat" : 55.7485

}

},

"@timestamp" : "2018-08-11T09:46:10.209Z",

"offset" : 0,

"tags" : [

"beats\_input\_codec\_plain\_applied"

],

"beat" : {

"version" : "6.3.2",

"hostname" : "localhost.localdomain",

"name" : "localhost.localdomain"

},

"clientip" : "83.149.9.216",

"@version" : "1",

"verb" : "GET",

"request" : "/presentations/logstash-monitorama-2013/images/kibana-search.png",

"prospector" : {

"type" : "log"

},

"referrer" : "\"http://semicomplete.com/presentations/logstash-monitorama-2013/\"",

"response" : "200",

"bytes" : "203023",

"source" : "/usr/local/programs/logstash/logstash-tutorial.log",

"agent" : "\"Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_9\_1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/32.0.1700.77 Safari/537.36\""

}

}

[IMG_277](https://www.cnblogs.com/cjsblog/p/javascript:void(0);)

再来一个

请求：

curl -XGET 'localhost:9200/logstash-2018.08.11/\_search?pretty&q=geoip.city\_name=Buffalo'

响应：

[IMG_278](https://www.cnblogs.com/cjsblog/p/javascript:void(0);)

{

"took" : 37,

"timed\_out" : false,

"\_shards" : {

"total" : 5,

"successful" : 5,

"skipped" : 0,

"failed" : 0

},

"hits" : {

"total" : 2,

"max\_score" : 2.6855774,

"hits" : [

{

"\_index" : "logstash-2018.08.11",

"\_type" : "doc",

"\_id" : "DvJhKGUBOuOlYJNtDPw7",

"\_score" : 2.6855774,

"\_source" : {

"host" : {

"name" : "localhost.localdomain"

},

"httpversion" : "1.1",

"ident" : "-",

"message" : "198.46.149.143 - - [04/Jan/2015:05:29:13 +0000] \"GET /blog/geekery/solving-good-or-bad-problems.html?utm\_source=feedburner&utm\_medium=feed&utm\_campaign=Feed%3A+semicomplete%2Fmain+%28semicomplete.com+-+Jordan+Sissel%29 HTTP/1.1\" 200 10756 \"-\" \"Tiny Tiny RSS/1.11 (http://tt-rss.org/)\"",

"auth" : "-",

"timestamp" : "04/Jan/2015:05:29:13 +0000",

"input" : {

"type" : "log"

},

"geoip" : {

"postal\_code" : "14202",

"region\_name" : "New York",

"timezone" : "America/New\_York",

"continent\_code" : "NA",

"city\_name" : "Buffalo",

"country\_code3" : "US",

"country\_name" : "United States",

"ip" : "198.46.149.143",

"dma\_code" : 514,

"country\_code2" : "US",

"region\_code" : "NY",

"latitude" : 42.8864,

"longitude" : -78.8781,

"location" : {

"lon" : -78.8781,

"lat" : 42.8864

}

},

"@timestamp" : "2018-08-11T09:46:10.254Z",

"offset" : 22795,

"tags" : [

"beats\_input\_codec\_plain\_applied"

],

"beat" : {

"version" : "6.3.2",

"hostname" : "localhost.localdomain",

"name" : "localhost.localdomain"

},

"clientip" : "198.46.149.143",

"@version" : "1",

"verb" : "GET",

"request" : "/blog/geekery/solving-good-or-bad-problems.html?utm\_source=feedburner&utm\_medium=feed&utm\_campaign=Feed%3A+semicomplete%2Fmain+%28semicomplete.com+-+Jordan+Sissel%29",

"prospector" : {

"type" : "log"

},

"referrer" : "\"-\"",

"response" : "200",

"bytes" : "10756",

"source" : "/usr/local/programs/logstash/logstash-tutorial.log",

"agent" : "\"Tiny Tiny RSS/1.11 (http://tt-rss.org/)\""

}

},

。。。

[IMG_279](https://www.cnblogs.com/cjsblog/p/javascript:void(0);)

（画外音：关于Elasticsearch可以查看另外一篇《[Elasticsearch 快速开始](https://www.cnblogs.com/cjsblog/p/9439331.html" \t "https://www.cnblogs.com/cjsblog/p/_blank)》）

**命令行启动Logstash**

为了从命令行启动Logstash，用下面的命令：

bin/logstash [options]

下面的例子展示了启动Logstash，并制定配置定义在mypipeline.conf文件中：

bin/logstash -f mypipeline.conf

在命令行中设置的任何标志都会覆盖**logstash.yml**中的相应设置。但是文件本身的内容没有改变。

**Command-Line Flags**

**--node.name NAME**

　　指定Logstash实例的名字。如果没有指定的话，默认是当前主机名。

**-f, --path.config CONFIG\_PATH**

　　从指定的文件或者目录加载Logstash配置。如果给定的是一个目录，则该目录中的所有文件将以字典顺序连接，然后作为一个配置文件进行解析。

**-e, --config.string CONFIG\_STRING**

用给定的字符串作为配置数据，语法和配置文件中是一样的。

**--modules**

运行的模块名字

**-l, --path.logs PATH**

Logstash内部日志输出目录

**--log.level LEVEL**

日志级别

**-t, --config.test\_and\_exit**

检查配置语法是否正确并退出

**-r, --config.reload.automatic**

监视配置文件的改变，并且当配置文件被修改以后自动重新加载配置文件。

**-config.reload.interval RELOAD\_INTERVAL**

为了检查配置文件是否改变，而拉去配置文件的频率。默认3秒。

**--http.host HTTP\_HOST**

Web API绑定的主机。REST端点绑定的地址。默认是"127.0.0.1"

**--http.port HTTP\_PORT**

Web API http端口。REST端点绑定的端口。默认是9600-9700之间。

**--log.format FORMAT**

指定Logstash写它自身的使用JSON格式还是文本格式。默认是"plain"。

**--path.settings SETTINGS\_DIR**

设置包含logstash.yml配置文件的目录，比如log4j日志配置。也可以设置LS\_SETTINGS\_DIR环境变量。默认的配置目录是在Logstash home目录下。

**-h, --help**

打印帮助

好了，关于Logstash介绍暂时先写这么多吧！ **感谢用心的自己**！！！

**其它相关**

《[Elasticsearch 快速开始](https://www.cnblogs.com/cjsblog/p/9439331.html" \t "https://www.cnblogs.com/cjsblog/p/_blank)》

《[开始使用Filebeat](https://www.cnblogs.com/cjsblog/p/9445792.html" \t "https://www.cnblogs.com/cjsblog/p/_blank)》