ELK指南

一、ELK概述

ELK 是三个开源软件的缩写，分别表示： Elasticsearch ， Logstash ， Kibana 。

ELK 通常用来构建日志分析平台、数据分析搜索平台等

官方文档

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

组件介绍

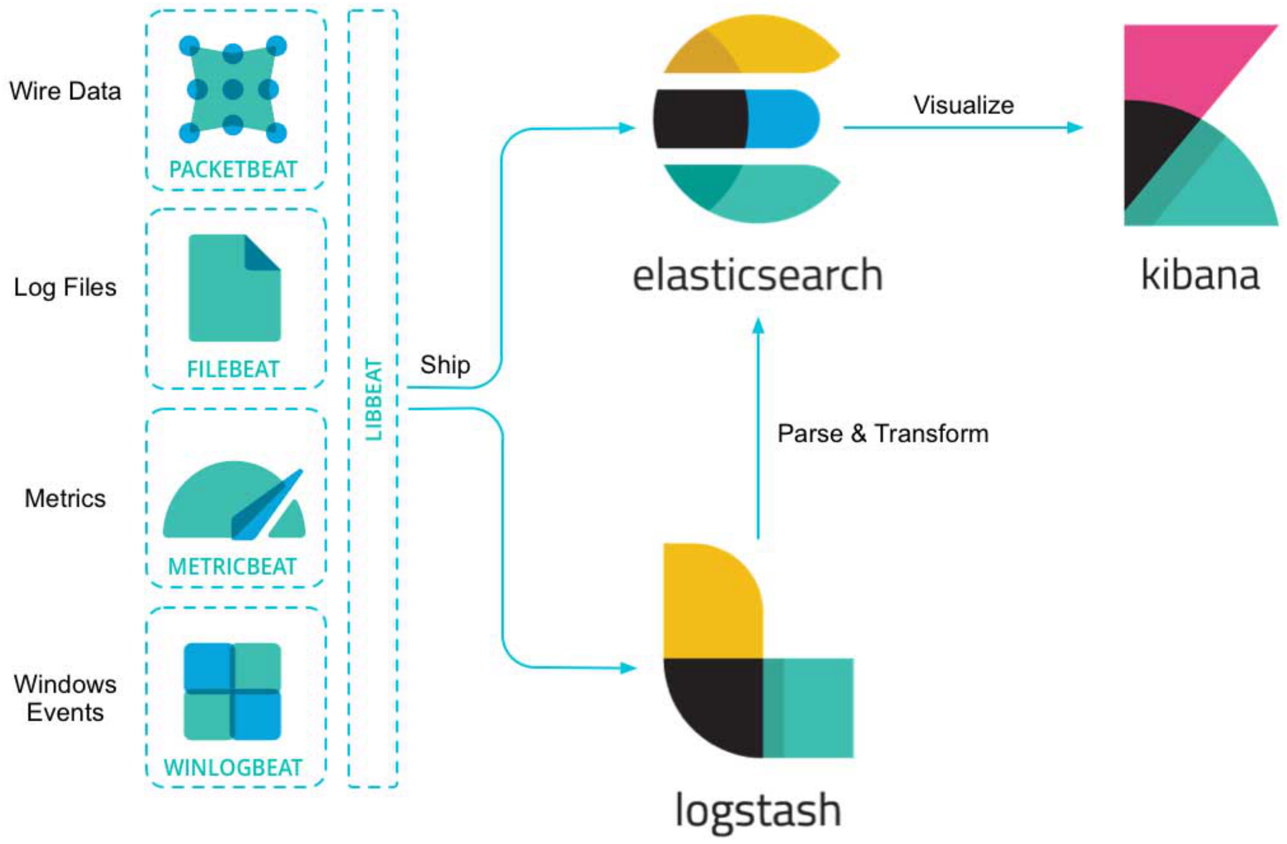
Elasticsearch 是个开源分布式全文检索和数据分析平台。它的特点有：分布式，零配置，自动发现，索引自动分片，索引副本机制，restful风格接口，负载均衡等特点。

Kibana 是一个针对Elasticsearch的开源数据分析及可视化平台，用来搜索、查看交互存储在Elasticsearch索引中的数据。使用Kibana，可以通过各种图表进行高级数据分析及展示。

Logstash 是一款基于插件的数据收集和处理引擎。Logstash 配有大量的插件，以便人们能够轻松进行配置以在多种不同的架构中收集、处理并转发数据。

Beats 轻量级的数据收集处理工具(Agent)，具有占用资源少的优点，适合于在各个服务器上采集数据后传输给Logstash，官方也推荐此工具。Beats有多多种类型，比较常用的是 FileBeats

组件关系图



二、Logstash详解

架构图

image-1-logstash-processing-pipeline.png

处理过程可分为一个或多个管道。在每个管道中，会有一个或多个 **输入插件** 接收或收集数据，然后这些数据会加入内部队列。默认情况下，这些数据很少并且会存储于内存中，但是为了提高可靠性和弹性，也可进行配置以 **扩大规模并长期存储在磁盘上**。

处理线程会以小批量的形式从队列中读取数据，并通过任何配置的 **过滤插件** 按顺序进行处理。Logstash 自带大量的插件，能够满足特定类型的操作需要，也就是解析、处理并丰富数据的过程。

处理完数据之后，处理线程会将数据发送到对应的输出插件，这些输出插件负责对数据进行格式化并进一步发送数据（例如发送到 Elasticsearch）。

准备工作

- 准备安装包 logstash-6.4.0.tar.gz

安装logstash

> **注意**：任何 Logstash 配置都必须至少包括一个输入插件和一个输出插件。过滤插件是可选项。

```
[root@localhost ~]# tar -zxvf logstash-6.4.0.tar.gz -C /usr
```

```
[root@localhost logstash-6.4.0]# cd /usr/logstash-6.4.0/
[root@localhost logstash-6.4.0]# vim config/simple.conf
# 第一个案例 配置内容如下
input {
 file {
   path => ["/root/testdata.log"]  # 需要采集数据的文件
   sincedb_path => "/dev/null"
   start_position => "beginning"   # 头开始读取文件
  }
}
filter {
}
output {
  stdout {
    codec => rubydebug
  }
}
```

启用测试

启动logstash服务

```
[root@localhost logstash-6.4.0]# bin/logstash -r -f config/simple.conf
Sending Logstash logs to /usr/logstash-6.4.0/logs which is now configured via log4j2.properties
[2019-01-05T22:00:57,538][INFO ][logstash.setting.writabledirectory] Creating directory
{:setting=>"path.queue", :path=>"/usr/logstash-6.4.0/data/queue"}
[2019-01-05T22:00:57,562][INFO ][logstash.setting.writabledirectory] Creating directory
{:setting=>"path.dead_letter_queue", :path=>"/usr/logstash-6.4.0/data/dead_letter_queue"}
[2019-01-05T22:01:01,781][WARN ][logstash.config.source.multilocal] Ignoring the
'pipelines.yml' file because modules or command line options are specified
[2019-01-05T22:01:02,043][INFO ][logstash.agent           ] No persistent UUID file found.
Generating new UUID {:uuid=>"603ad236-368d-45c1-bb93-40f33d8a794c", :path=>"/usr/logstash-
6.4.0/data/uuid"}
[2019-01-05T22:01:04,194][INFO ][logstash.runner          ] Starting Logstash
{"logstash.version"=>"6.4.0"}
[2019-01-05T22:01:14,944][INFO ][logstash.pipeline        ] Starting pipeline
{:pipeline_id=>"main", "pipeline.workers"=>4, "pipeline.batch.size"=>125,
"pipeline.batch.delay"=>50}
[2019-01-05T22:01:31,847][INFO ][logstash.pipeline        ] Pipeline started successfully
{:pipeline_id=>"main", :thread=>"#<Thread:0x170cd118 run>"}
[2019-01-05T22:01:31,959][INFO ][logstash.agent           ] Pipelines running {:count=>1,
:running_pipelines=>[:main], :non_running_pipelines=>[]}
[2019-01-05T22:01:31,990][INFO ][filewatch.observingtail  ] START, creating Discoverer, Watch
with file and sincedb collections
[2019-01-05T22:01:32,734][INFO ][logstash.agent           ] Successfully started Logstash API
endpoint {:port=>9600}
```

新建数据文件保存退出

```
[root@localhost ~]# vim testdata.log

# 输入内容
Hello Logstash
```

logstash向控制台提供的输出显示

```
{
    "@timestamp" => 2019-01-06T03:04:18.645Z,
          "path" => "/root/testdata.log",
       "message" => "Hello Logstash",
      "@version" => "1",
          "host" => "localhost.localdomain"
}
```

三、使用Logstash解析Nginx日志

安装Nginx

略

Nginx访问日志

`access.log` 数据如下:

```
192.168.23.1 - - [07/Jan/2019:03:38:21 -0500] "GET /favicon.ico HTTP/1.1" 404 571
"http://192.168.23.143/" "Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/71.0.3578.80 Safari/537.36"
```

已第一条记录为例，其中

- `192.168.23.1` 客户端地址
- `-` 客户端用户名
- `[07/Jan/2019:03:38:21 -0500]` 服务器时间
- `GET /favicon.ico HTTP/1.1` 请求内容，包括方法名，地址，和http协议
- `404` 返回的http 状态码
- `571` 返回的大小
- `http://192.168.23.143/` 可以记录用户是从哪个链接访问过来的
- `Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/71.0.3578.80 Safari/537.36` 用户所使用的代理（一般为浏览器）

使用Grok插件解析数据

> Grok插件用来将非结构化的数据解析为结构化数据

参考资料： Filter plugins » grok

你还需要以下这个网站对你所编写的过滤匹配代码进行debug：

grok debuger

上面的日志的过滤代码如下：

```
%{IPORHOST:client_ip} - %{USER:auth} \[%{HTTPDATE:timestamp}\] "(?:%{WORD:verb} %
{NOTSPACE:request}(?: HTTP/%{NUMBER:httpversion})?|%{DATA:rawrequest})" %{NUMBER:response} (?:%
{NUMBER:bytes}|-) %{QS:http_referer} %{QS:http_user_agent}
```

更多grok语法请参考：https://github.com/logstash-plugins/logstash-patterns-core/blob/master/patterns/grok-patterns

测试

修改 `logstash` 配置文件,内容如下：

```
input {
 file {
   path => ["/usr/local/nginx/logs/access.log"]
   sincedb_path => "/dev/null"
   start_position => "beginning"
  }
}
filter {
  grok {
    match =>{
        "message" => "%{IPORHOST:client_ip} - %{USER:auth} \[%{HTTPDATE:timestamp}\] \"(?:%
{WORD:verb} %{NOTSPACE:request}(?: HTTP/%{NUMBER:httpversion})?|%{DATA:rawrequest})\" %
{NUMBER:response} (?:%{NUMBER:bytes}|-) %{QS:http_referer} %{QS:http_user_agent}"
    }
  }
  geoip {
      source => "client_ip"
  }

  date {
      match => [ "time" , "dd/MMM/YYYY:HH:mm:ss Z" ]
   }
}
output {
  stdout {
    codec => rubydebug
  }
}
```

> - geoip：使用GeoIP数据库对client_ip字段的IP地址进行解析，可得出该IP的经纬度、国家与城市等信息，但精确度不高，这主要依赖于GeoIP数据库；
> - date：默认情况下，elasticsearch内记录的date字段是elasticsearch接收到该日志的时间，但在实际应用中需要修改为日志中所记录的时间。这时候则需要指定记录时间的字段并指定时间格式。如果匹配成功，则会将日志的时间替换至date字段中。

logstash向控制台提供的输出显示

```
{
          "timestamp" => "07/Jan/2019:05:08:38 -0500",
           "response" => "304",
    "http_user_agent" => "\"Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/71.0.3578.98 Safari/537.36\"",
               "auth" => "-",
        "httpversion" => "1.1",
          "client_ip" => "110.52.250.126",
            "message" => "110.52.250.126 - - [07/Jan/2019:05:08:38 -0500] \"GET / HTTP/1.1\" 304 0 \"-\" \"Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/71.0.3578.98 Safari/537.36\"",
              "geoip" => {
         "country_code2" => "CN",
             "city_name" => "Changsha",
              "latitude" => 28.1792,
              "location" => {
            "lon" => 113.1136,
            "lat" => 28.1792
        },
          "country_name" => "China",
              "timezone" => "Asia/Shanghai",
             "longitude" => 113.1136,
        "continent_code" => "AS",
           "region_code" => "43",
                    "ip" => "110.52.250.126",
         "country_code3" => "CN",
           "region_name" => "Hunan"
    },
               "verb" => "GET",
               "host" => "MiWiFi-R3P-srv",
         "@timestamp" => 2019-01-07T10:22:39.711Z,
              "bytes" => "0",
               "path" => "/usr/local/nginx/logs/access.log",
       "http_referer" => "\"-\"",
           "@version" => "1",
            "request" => "/"
}
```

四、使用ELK搭建日志采集分析平台

Filebeat环境搭建

安装

```
[root@localhost ~]# tar -zxvf filebeat-6.4.0-linux-x86_64.tar.gz -C /usr
```

配置

```
[root@localhost ~]# mkdir logs
[root@localhost ~]# vim /usr/filebeat-6.4.0-linux-x86_64/filebeat.yml

- type: log
```

```
  # Change to true to enable this input configuration.
  enabled: true

  # Paths that should be crawled and fetched. Glob based paths.
  paths:
    - /usr/local/nginx/logs/access*.log
# output.elasticsearch:
  # Array of hosts to connect to.
  # hosts: ["localhost:9200"]
output.logstash:
  # The Logstash hosts
  hosts: ["192.168.23.143:5044"]
```

上传测试数据

略

Logstash环境搭建

配置文件

```
# 配置输入为 beats
input {
  beats {
    port => "5044"
  }
}
# 数据过滤 解析
filter {
  grok {
    match =>{
        "message" => "%{IPORHOST:client_ip} - %{USER:auth} \[%{HTTPDATE:timestamp}\] \"(?:%
{WORD:verb} %{NOTSPACE:request}(?: HTTP/%{NUMBER:httpversion})?|%{DATA:rawrequest})\" %
{NUMBER:response} (?:%{NUMBER:bytes}|-) %{QS:http_referer} %{QS:http_user_agent}"
    }
  }
  geoip {
      source => "client_ip"
  }

  date {
      match => [ "time" , "dd/MMM/YYYY:HH:mm:ss Z" ]
   }
}
# 输出到本机的 ES
output {
  elasticsearch {
      hosts => [ "192.168.23.143:9200"  ]
      index => "logs-%{+YYYY.MM.dd}"
  }
}
```

启动服务测试

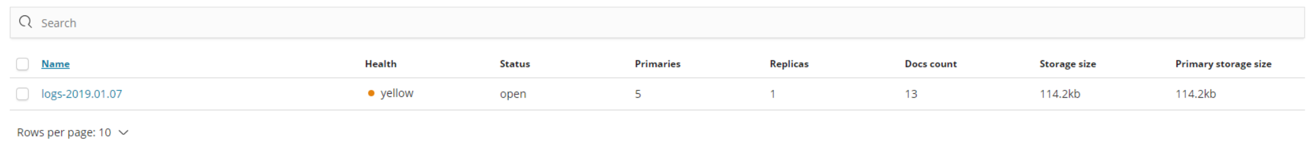
启动logstash

```
[root@localhost logstash-6.4.0]# bin/logstash -r -f config/elk.conf
```

启动filebeat

```
[root@localhost filebeat-6.4.0-linux-x86_64]# ./filebeat
```

数据采集

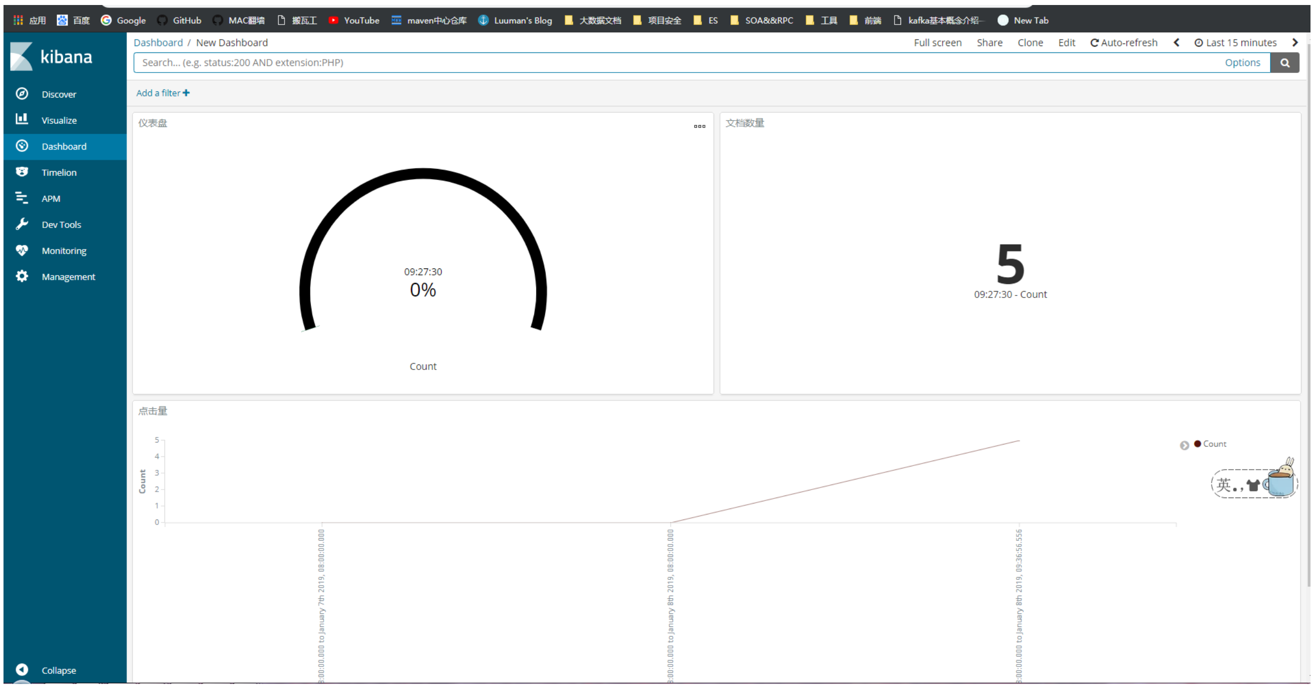


Search

	Name	Health	Status	Primaries	Replicas	Docs count	Storage size	Primary storage size
☐	logs-2019.01.07	● yellow	open	5	1	13	114.2kb	114.2kb

Rows per page: 10

Kibana数据可视化展示



五、构建ES集群

elasticsearch集群的搭建特别的简单：

1. 在集群的每个节点上，将elasticsearch的单点安装好
2. **修改配置文件elasticsearch.yml的cluster.name**(集群名称)配置，要求所有节点配置一致
3. 修改配置文件elasticsearch.yml的http.port为9200，所有节点配置统一

配置文件修改内容如下：

```
cluster.name: es-cluster
node.name: node-1  # 另外一节点为：node-2
network.host: 192.168.23.141  # 另外一节点为 network.host: 192.168.23.141
discovery.zen.ping.unicast.hosts: ["192.168.23.141:9300", "192.168.23.142:9300"]
```

启动服务测试

```
GET /_cat/nodes?v

ip             heap.percent ram.percent cpu load_1m load_5m load_15m node.role master name
192.168.23.142           32          94   8    0.28    0.34     0.25 mdi       *      node-2
192.168.23.141           28          96  13    0.54    0.59     0.54 mdi       -      node-1
```

六、深度分页解决方案

https://es.xiaoleilu.com/060_Distributed_Search/20_Scan_and_scroll.html