# 0.介绍

logstash体系结构:源数据源>输入编解码插件>事件队列>过滤器插件>输出编解码插件>目标数据源

# 1.安装

## 1.1.解压方式

### 1.1.1. 解压

tar -xvf logstash-7.7.0-linux-x86\_64.tar.gz

# 2.bin

## 2.1. 显示已安装了的插件

./bin/logstash-plugin list

## **2.2. 显示插件命令帮助信息**

./bin/logstash-plugin --help

## **2.3. logstash命令行方式**

示例1：从控制台输入，由控制台输出

./bin/logstash -e "input{stdin{}} output{stdout{}}"

示例2：从控制台输入，由控制台输出，设置输出的编解码器

codec=plain

./bin/logstash -e "input{stdin{}} output{stdout{codec=plain}}"

## **2.4. conf文件方式**

### 2.4.1. conf文件内容

input{

stdin{

}

}

output {

elasticsearch{

hosts => ["http://ip:9200"]

index => "mystdin"

#user => "elastic"

#password => "changeme"

}

}

### **2.4.2. 执行logstash命令**

./logstash -f std\_es.conf

### **2.4.3 kibana中查询结果**

get /mystdin/\_search

### **2.4.3 校验conf文件的语法**

./logstash -f xxxxxxx.conf -t

# 插件

## 3.1.编解码插件

### 3.1.1.plain插件

### 3.1.2.line插件

### 3.1.3.json插件

### 3.1.4.序列化插件

## **3.2.输入输出插件**

### 3.2.1.stdin输入插件和stout输出插件

### 3.2.2.elasticsearch插件

### 3.2.3.文件插件

范例1：

input{

file{

path => “home/elk/elk/teaching/elasticsearch-7.7.0/logs/my-elk.log”

start\_position => “beginning” #[beginning,ending]

codec => multiline { #只要不是’[’开头的行都向前合并

pattern => “^\[”

begate => true

what => “previous”

}

}

}

output {

elasticsearch{

hosts => ["http://ip:9200"]

index => "es-log-%{+YYYY.MM.dd}" #对应的日志文件也做类似的分片管理，每天的日志存一个文件

#user => "elastic"

#password => "changeme"

}

}

## **3.3.过滤器插件**

### 3.3.1.全文数据结构化过滤器

范例1：

input {

file {

path => "/home/elk/elk/teaching/elasticsearch-7.7.0/logs/my-elk.log"

start\_position => "beginning"

codec => multiline {

pattern => "^\["

negate => true

what => "previous"

}

}

}

filter{

grok {

match => {

message => "\[%{TIMESTAMP\_ISO8601:time}\]\[%{LOGLEVEL:level}%{SPACE}\]\[%{NOT

SPACE:loggerclass}%{SPACE}\]%{SPACE}\[%{DATA:nodename}\]%{SPACE}

%{GREEDYDATA:msg}"

} #通过grok插件将元数据message分解成子数据，TIMESTAMP\_ISO8601等是预定义的格式

}

}

output {

stdout{}

}

范例2：

input {

file {

path => "/home/elk/elk/teaching/elasticsearch-7.7.0/logs/my-elk.log"

start\_position => "beginning"

codec => multiline {

pattern => "^\["

negate => true

what => "previous"

}

}

}

filter{

grok {

match => {

message => "\[%{TIMESTAMP\_ISO8601:time}\]\[%{LOGLEVEL:level}%{SPACE}\]\[%{NOT

SPACE:loggerclass}%{SPACE}\]%{SPACE}\[%{DATA:nodename}\]%{SPACE}

%{GREEDYDATA:msg}"

} #通过grok插件将元数据message分解成子数据，TIMESTAMP\_ISO8601等是预定义的格式

}

}

output {

elasticsearch {

hosts => ["http://ip:9200"]

index => "es-log-text-%{+YYYY.MM.dd}"

template\_name => "es\_template\*" #模版文件名称

template => "/home/elk/elk/teaching/logstash-7.7.0/config" #模版文件所在文件夹路径

}

stdout{}

}

config目录下的es\_template.json文件



### 3.3.2.处理半结构化文本

# Springboot集成logstash

https://zhuanlan.zhihu.com/p/140883168