# 原 Logstash配置总结和实例

shawnplaying 发表于 2年前 阅读 9661 收藏 23 点赞 3 评论 0



*摘要: Logstash配置总结和实例*

这里记录Logstash配置中注意的事项：

整个配置文件分为三部分：input,filter,output。参考这里的介绍 https://www.elastic.co/guide/en/logstash/current/configuration-file-structure.html

1 在Windows中，文件路径中分隔符要使用/而不是\。如果使用了\，那么*匹配将会失败。

2 默认的@timestamp是使用UTC时间表示的，所以对于北京时间会有8小时差，我们很多情况会使用日期来做type区分日志，这时有个办法是在filter中增加ruby来做转换。其实是新加了一个属性来表示日期。

另外有个选项是在date中有timezone的属性，但是如果设置它为UTC，那么整个日志中的时间就是UTC时间了，不是一个理想的做法。

下面是一个配置解释

2016-5-5更新：

```
#整个配置文件分为三部分：input,filter,output
#参考这里的介绍 https://www.elastic.co/guide/en/logstash/current/configuration-file-structure.html
input {
  #file可以多次使用，也可以只写一个file而设置它的path属性配置多个文件实现多文件监控
  file {
    #type是给结果增加了一个属性叫type值为"<xxx>"的条目。这里的type，对应了ES中index中的type，即如果输入ES时，没有指定
    type => "apache-access"
    path => "/apphome/ptc/Windchill_10.0/Apache/logs/access_log*"
    #start_position可以设置为beginning或者end，beginning表示从头开始读取文件，end表示读取最新的，这个也要和ignore_ol
    start_position => beginning
    #sincedb_path表示文件读取进度的记录，每行表示一个文件，每行有两个数字，第一个表示文件的inode，第二个表示文件读取到
    sincedb_path => "/opt/logstash-2.3.1/sincedb_path/access_progress"
    #ignore_older表示了针对多久的文件进行监控，默认一天，单位为秒，可以自己定制，比如默认只读取一天内被修改的文件。
    ignore_older => 604800
    #add_field增加属性。这里使用了${HOSTNAME}，即本机的环境变量，如果要使用本机的环境变量，那么需要在启动命令上加--alo
    add_field => {"log_hostname"=>"${HOSTNAME}"}
    #这个值默认是\n 换行符，如果设置为空""，那么后果是每个字符代表一个event
    delimiter => ""
    #这个表示关闭超过（默认）3600秒后追踪文件。这个对于multiline来说特别有用。... 这个参数和logstash对文件的读取方式有
    close_older => 3600
    coodec => multiline {
      pattern => "^\s"
      #这个negate是否定的意思，意思跟pattern相反，也就是不满足patter的意思。
#      negate => ""
      #what有两个值可选 previous和next，举例说明，java的异常从第二行以空格开始，这里就可以pattern匹配空格开始，what设
      what => "previous"
      auto_flush_interval => 60
    }
  }
  file {
    type => "methodserver-log"
    path => "/apphome/ptc/Windchill_10.0/Windchill/logs/MethodServer-1604221021-32380.log"
    start_position => beginning
    sincedb_path => "/opt/logstash-2.3.1/sincedb_path/methodserver_process"
    ignore_older => 604800
  }
```

执行ruby程序，下面例子是将日期转化为字符串赋予daytag

```
  ruby {
    code => "event['daytag'] = event.timestamp.time.localtime.strftime('%Y-%m-%d')"
  }
  # if [path] =~ "access" {} else if [path] =~ "methodserver" {} else if [path] =~ "servermanager" {} else {}
  if [path] =~ "MethodServer" { #z这里的=~是匹配正则表达式
    grok {
      patterns_dir => ["/opt/logstash-2.3.1/patterns"] #自定义正则匹配
#       Tue 4/12/16 14:24:17: TP-Processor2: hirecode---->77LS
      match => { "message" => "%{DAY:log_weekday} %{DATE_US:log_date} %{TIME:log_time}: %{GREEDYDATA:log_data}
    }
    #mutage是做转换用的
    mutate {
      replace => { "type" => "apache" } #替换属性值
      convert => { #类型转换
        "bytes" => "integer" #例如还有float
        "duration" => "integer"
        "state" => "integer"
      }
    #date主要是用来处理文件内容中的日期的。内容中读取的是字符串，通过date将它转换为@timestamp。参考https://www.elasti
#    date {
#      match => [ "logTime" , "dd/MMM/yyyy:HH:mm:ss Z" ]
#    }
  }else if [type] in ['tbg_qas','mbg_pre'] { # if ... else if ... else if ... else结构
  }else {
    drop{} # 将event丢弃
  }
}
output {
  stdout{ codec=>rubydebug} # 直接输出，调试用起来方便
  # 输出到redis
  redis {
    host => '10.120.20.208'
    data_type => 'list'
    key => '10.99.201.34:access_log_2016-04'
  }
  # 输出到ES
  elasticsearch {
    hosts =>"192.168.0.15:9200"
    index => "%{sysid}_%{type}"
    document_type => "%{daytag}"
  }
}
```

下面是两个真实的例子，第一个是从应用到redis，第二个是从redis到ES。

```
input {
  file {
    type => "log_raw_data"
    path => "/apphome/ptc/Windchill_10.0/Windchill/logs/gc/*GC.log"
    start_position => end
    sincedb_path => "/opt/logstash-2.3.1/sincedb_path/log_progress"
#    ignore_older => 604800
    add_field => {"sysid"=>"tbg_qas"}
  }
  file {
    type => "log_raw_data"
    path => ["/apphome/ptc/Windchill_10.0/Windchill/logs/*MethodServer*.log","/apphome/ptc/Windchill_10.0/Wind
    start_position => end
    sincedb_path => "/opt/logstash-2.3.1/sincedb_path/log_progress"
#    ignore_older => 604800
    add_field => {"sysid"=>"tbg_qas"}
    close_older => 60
    codec => multiline {
#      patterns_dir => ["D:/app/logstash-2.3.1/patterns"]
      pattern => "^%{DAY} %{DATESTAMP}:"
      negate => true
      what => "previous"
#      auto_flush_interval => 20
    }
  }
}
output {
#  stdout{ codec=>rubydebug}
  redis {
    host => '10.120.20.208'
    data_type => 'list'
    key => 'log_raw_data'
  }
  redis {
    host => '10.120.31.142'
    data_type => 'list'
    key => 'log_raw_data'
  }
}
```

```
input {
  redis {
    host => "localhost"
    data_type => "list"
    port => "6379"
    key => "log_raw_data"
    type => "redis-input"
  }
}
filter{
  ruby {
    code => "event['daytag'] = event.timestamp.time.localtime.strftime('%Y-%m-%d')"
  }
  if [path] =~ "access" {
    grok {
      match => { "message" => "%{IPORHOST:clientip} %{HTTPDUSER:ident} %{USER:username} \[%{HTTPDATE:logtime}\
    }
    mutate {
      replace => { "type" => "apache" }
      convert => {
        "bytes" => "integer"
        "duration" => "integer"
        "state" => "integer"
      }
    }
    date {
      match => [ "logtime" , "dd/MMM/yyyy:HH:mm:ss Z" ]
    }
  }else if [path] =~ ".*ServerManager.*GC\.log" {
    if [message] =~ "\[Full GC" {
      grok {
        match => {"message" => "%{TIMESTAMP_ISO8601:logtime}: %{GREEDYDATA:gcdetail} \[Times: user=%{BASE10NUM
      }
      date {
        match => ["logtime" , "yyyy-MM-dd'T'HH:mm:ss.SSS'+0800'"]
      }
    }else if [message] =~ "\[GC" {
      grok {
        match => {"message" => "%{TIMESTAMP_ISO8601:logtime}: %{GREEDYDATA:gcdetail} \[Times: user=%{BASE10NUM
      }
      date {
        match => ["logtime" , "yyyy-MM-dd'T'HH:mm:ss.SSS'+0800'"]
      }
    }else{
      drop {}
    }
    mutate {
      replace => {"type" => "smgc" }
      convert => {
        "usertime" => "float"
        "systime" => "float"
        "realtime" => "float"
      }
    }
  }else if [path] =~ ".*MethodServer.*GC\.log" {
    if [message] =~ "\[Full GC" {
      grok {
        match => {"message" => "%{TIMESTAMP_ISO8601:logtime}: %{GREEDYDATA:gcdetail} \[Times: user=%{BASE10NUM
      }
      date {
        match => ["logtime" , "yyyy-MM-dd'T'HH:mm:ss.SSS'+0800'"]
      }
    }else if [message] =~ "\[GC" {
      grok {
        match => {"message" => "%{TIMESTAMP_ISO8601:logtime}: %{GREEDYDATA:gcdetail} \[Times: user=%{BASE10NUM
      }
      date {
        match => ["logtime" , "yyyy-MM-dd'T'HH:mm:ss.SSS'+0800'"]
      }
    }else{
      drop {}
    }
    mutate {
      replace => {"type" => "msgc" }
      convert => {
        "usertime" => "float"
        "systime" => "float"
        "realtime" => "float"
      }
    }
  }else if [path] =~ "MethodServer" {
    grok {
      match => { "message" => "%{DAY:weekday} %{DATESTAMP:logtime}: %{GREEDYDATA:logdata}"}
    }
```

```
    date {
      match => [ "logtime" , "M/d/yy HH:mm:ss" ]
    }
    mutate { replace => { "type" => "ms" } }
  }else if [path] =~ "ServerManager" {
    grok {
      match => { "message" => "%{DAY:weekday} %{DATESTAMP:logtime}: %{GREEDYDATA:logdata}"}
    }
    date {
      match => [ "logtime" , "M/d/yy HH:mm:ss" ]
    }
    mutate { replace => { "type" => "sm" } }
  }else if [path] =~ "Process_Archive" {
    grok {
      patterns_dir => ["/opt/logstash-2.3.1/patterns"]
      match => { "message" => "%{PROCESS_DATETIME:logtime} %{GREEDYDATA:logdata}"}
    }
    date {
      match => [ "logtime" , "yyyy MMM dd HH:mm:ss:SSS 'GMT +8'" ]
    }
    mutate { replace => { "type" => "prc_arc" } }
  }else if [path] =~ "ESISAPAdapterConfiguration" {
    grok {
      patterns_dir => ["/opt/logstash-2.3.1/patterns"]
      match => { "message" => "%{PROCESS_DATETIME:logtime} %{GREEDYDATA:logdata}"}
    }
    date {
      match => [ "logtime" , "yyyy MMM dd HH:mm:ss:SSS 'GMT +8'" ]
    }
    mutate { replace => { "type" => "esi_adp" } }
  }else if [path] =~ "LenovoAdapterConfiguration" {
    grok {
      patterns_dir => ["/opt/logstash-2.3.1/patterns"]
      match => { "message" => "%{PROCESS_DATETIME:logtime} %{GREEDYDATA:logdata}"}
    }
    date {
      match => [ "logtime" , "yyyy MMM dd HH:mm:ss:SSS 'GMT +8'" ]
    }
    mutate { replace => { "type" => "le_adp" } }
  }else {
    mutate { replace => { "type" => "other" } }
#    drop {}
  }
#  extractnumbers {
#    source => "duration"
#  }
}
output {
#  stdout{ codec=>rubydebug}
  elasticsearch {
    hosts =>"192.168.0.15:9200"
    index => "%{sysid}_%{type}"
    document_type => "%{daytag}"
  }
}
```



分类：检索　字数：1572

标签： Logstash



打赏　点赞　收藏　分享

举报

shawnplaying

系统管理员 海淀

+ 关注

粉丝 14 | 博文 125 | 码字总数 70640

## 相关博客

Logstash详解

张欢19933

347 0

logstash 配置

流萤飘枫丶

76 0

logstash实战

ville

117 0

## 评论 (0)

Ctrl+Enter 发表评论