在logstash与elasticsearch集成的时候，总共有如下几种使用模板的方式：   
  
（1）使用默认自带的索引模板 ，大部分的字段都会分词，适合开发和时候快速验证使用   
（2）在logstash收集端自定义配置模板，因为分散在收集机器上，维护比较麻烦   
（3）在elasticsearc服务端自定义配置模板，由elasticsearch负责加载模板，可动态更改，全局生效，维护比较容易   
  
以上几种方式：   
  
使用第一种，最简单，无须任何配置   
使用第二种，适合小规模集群的日志收集，需要在logstash的output插件中使用template指定本机器上的一个模板json路径， 例如  template => "/tmp/logstash.json"   
使用第三种，适合大规模集群的日志收集，如何配置，主要配置logstash的output插件中两个参数： 

**Java代码  [收藏代码](javascript:void())**

1. manage\_template => **false**//关闭logstash自动管理模板功能
2. template\_name => "crawl"//映射模板的名字

命令：

put \_template/template\_name

{

\*\*\*\*

}  
如果使用了，第三种需要在elasticsearch的集群中的config/templates路径下配置模板json，在elasticsearch中索引模板可分为两种：   
  
（一）：静态模板   
  
适合索引字段数据固定的场景，一旦配置完成，不能向里面加入多余的字段，否则会报错   
  
优点：scheam已知，业务场景明确，不容易出现因字段随便映射从而造成元数据撑爆es内存，从而导致es集群全部宕机   
缺点：字段数多的情况下配置稍繁琐   
  
一个静态索引模板配置例子如下：

{

"logstash" : {

"template": "logstash-\*",

"settings": {

"index.number\_of\_shards": 5,

"number\_of\_replicas": 0

},

"mappings" : {

"logs" : {

"properties" : {

"@timestamp": {

"include\_in\_all": false,

"type": "date"

},

"advId" : {

"type" : "long"

},

"appid" : {

"type" : "string",

},

"callAdvTime" : {

"type" : "date",

},

"callAdvUrl" : {

"type" : "string",

},

"channelTime" : {

"type" : "date",

},

"client\_ip" : {

"type" : "ip",

},

"delMode" : {

"type" : "string",

},

"delPlat" : {

"type" : "string",

},

"from\_ip" : {

"type" : "ip",

},

"host" : {

"type" : "string",

},

"landingPageId" : {

"type" : "long",

},

"level" : {

"type" : "string",

},

"level\_value" : {

"type" : "long"

},

"log\_type" : {

"type" : "string",

},

"logger\_name" : {

"type" : "string",

},

"message" : {

"type" : "string",

},

"orderId" : {

"type" : "long"

},

"port" : {

"type" : "long"

},

"proId" : {

"type" : "long"

},

"repeatTime" : {

"type" : "long"

},

"responseMsg" : {

"type" : "string",

},

"scid" : {

"type" : "string",

},

"source" : {

"type" : "string",

},

"sourceId" : {

"type" : "long"

},

"status" : {

"type" : "long"

},

"tags" : {

"type" : "string",

},

"thread\_name" : {

"type" : "string",

},

"uuid" : {

"type" : "string",

}

}

}

}

二、动态模板

{

"joy\_dm2": {

"aliases": {},

"mappings": {

"logs": {

"dynamic\_templates": [

{

"date\_as\_date": {

"match\_mapping\_type": "date",

"mapping": {

"type": "date"

}

}

},

{

"Time": {

"match": "\*Time",

"unmatch": "repeatTime",

"match\_mapping\_type": "string",

"mapping": {

"type": "date"

}

}

},

{

"ip\_as\_ip": {

"match": "\*ip",

"match\_mapping\_type": "string",

"mapping": {

"type": "ip"

}

}

},

{

"url\_as\_text": {

"match": "\*Url",

"match\_mapping\_type": "string",

"mapping": {

"type": "text"

}

}

},

{

"string\_as\_text": {

"match\_mapping\_type": "string",

"mapping": {

"type": "keyword"

}

}

}

]

}

},

"settings": {

"index": {

"codec": "best\_compression",

"number\_of\_shards": "5",

"number\_of\_replicas": "1"

}

}

}

}