Logstash 参考指南（Logstash工作原理）

Logstash工作原理

Logstash事件处理管道有三个阶段：输入→过滤器→输出，输入生成事件，过滤器修改它们，然后输出将它们发送到其他地方。输入和输出支持编解码器，使你能够在数据进入或离开管道时对其进行编码或解码，而无需使用单独的过滤器。

输入

你使用输入将数据获取到Logstash中，一些比较常用的输入是：

- **file**：从文件系统上的文件进行读取，非常类似于UNIX命令`tail -0F`。
- **syslog**：在众所周知的端口514上监听syslog消息并根据RFC3164格式进行解析。
- **redis**：从redis服务器读取数据，同时使用Redis通道和Redis列表，Redis通常被用作集中式Logstash安装中的“broker”，它将从远程Logstash “shipper”中的Logstash事件排队。
- **beats**：处理Beats发送的事件。

有关可用输入的更多信息，请参见输入插件。

过滤器

过滤器是Logstash管道中的中间处理设备，如果事件符合一定的条件，你可以将过滤器与条件语句组合在一起，对其执行操作，一些有用的过滤器包括：

- **grok**：解析和构造任意文本，Grok是目前Logstash中解析非结构化日志数据到结构化和可查询数据的最佳方式，使用内置的120种模

式，你很可能会找到一个满足你的需要！
- **mutate**：对事件字段执行一般的转换，你可以重命名、删除、替换和修改事件中的字段。
- **drop**：完全删除事件，例如debug事件。
- **clone**：复制事件，可能添加或删除字段。
- **geoip**：添加关于IP地址地理位置的信息（在Kibana中还显示了令人惊叹的图表！）

有关可用过滤器的更多信息，请参见过滤器插件。

输出

输出是Logstash管道的最后阶段，事件可以通过多个输出，但是一旦所有的输出处理完成，事件就完成了它的执行，一些常用的输出包括：

- **elasticsearch**：发送事件数据到Elasticsearch，如果你打算以一种高效、方便、易于查询的格式保存数据，那么使用Elasticsearch是可行的。
- **file**：将事件数据写入磁盘上的文件。
- **graphite**：将事件数据发送到graphite，这是一种流行的用于存储和绘制指标的开源工具。http://graphite.readthedocs.io/en/latest/
- **statsd**：发送事件到statsd，“监听统计信息（如计数器和计时器）、通过UDP发送聚合并将聚合发送到一个或多个可插拔后端服务”的服务，如果你已经在使用statsd，这可能对你很有用！

有关可用输出的更多信息，请参见输出插件。

编解码器

Codecs是基本的流过滤器，可以作为输入或输出的一部分进行操作，Codecs使你能够轻松地将消息的传输与序列化过程分开，流行的codecs包括`json`、`msgpack`和`plain`（text）。

- **json**：以JSON格式对数据进行编码或解码。

- **multiline**：将多行文本事件（如java异常和stacktrace消息）合并到单个事件中。

有关可用编解码器的更多信息，请参见编解码器插件。

执行模型

Logstash事件处理管道协调输入、过滤器和输出的执行。

Logstash管道中的每个输入阶段都在自己的线程中运行，输入将事件写入位于内存（默认）或磁盘上的中央队列，每个管道工作线程从这个队列中取出一批事件，通过配置的过滤器运行事件批处理，然后通过任何输出运行过滤的事件，可以配置批处理的大小和管道工作线程的数量（参见调优和分析*Logstash*性能）。

默认情况下，Logstash使用内存有限队列在管道阶段之间（输入→过滤器和过滤器→输出）来缓冲事件，如果Logstash不安全的终止，则存储在内存中的任何事件都将丢失。为了防止数据丢失，你可以启用Logstash将运行中的事件持久化到磁盘上，有关更多信息，请参见持久队列。