

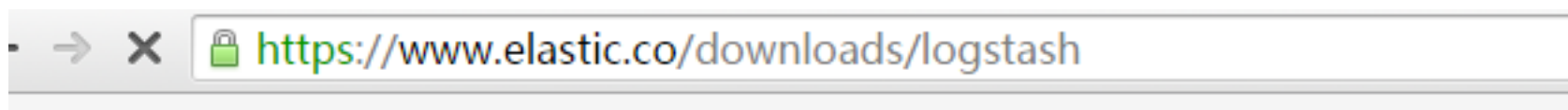
# [Logstash]使用详解

Logstash是一款轻量级的日志搜集处理框架，可以方便的把分散的、多样化的日志搜集起来，并进行自定义的处理，然后传输到指定的位置，比如某个服务器或者文件。

本文针对[官方文档](#)进行翻译以及实践，希望有更多的有用户了解、使用这款工具。

## 下载、安装、使用

这款工具是开箱即用的软件，[下载地址戳这里](#)，下载自己对应的系统版本即可。



### Logstash 1.5.4

ZIP sha1

TAR.GZ sha1

DEB sha1

RPM sha1

下载后直接解压，就可以了。

通过命令行，进入到logstash/bin目录，执行下面的命令：

可以看到提示下面信息（这个命令稍后介绍），输入**hello world!**

```
管理员: C:\Windows\system32\cmd.exe - logstash -e ""

E:\software\logstash-1.5.4\logstash-1.5.4\bin>logstash -e ""
io/console not supported; tty will not be manipulated
Logstash startup completed
hello world!
<
  "message" => "hello world!\r",
  "@version" => "1",
  "@timestamp" => "2015-09-12T04:09:38.130Z",
  "type" => "stdin",
  "host" => "PC-201507311411"
>
```

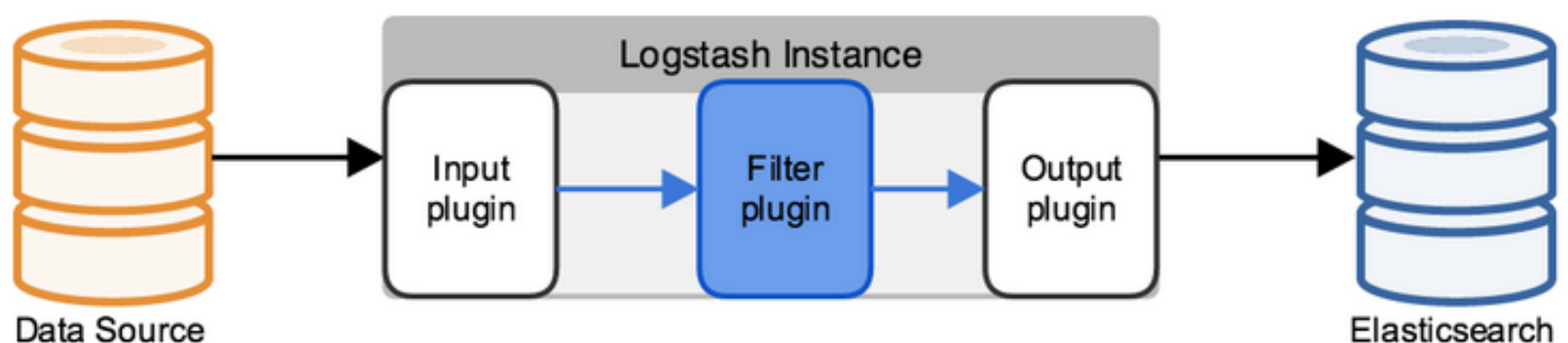
可以看到logstash尾我们自动添加了几个字段，时间戳@timestamp，版本@version，输入的类型type，以及主机名host。

## 工作原理

Logstash使用管道方式进行日志的搜集处理和输出。有点类似\*NIX系统的管道命令 **xxx | ccc | ddd**，xxx执行完了会执行ccc，然后执行ddd。

在logstash中，包括了三个阶段：

输入input --> 处理filter（不是必须的） --> 输出output



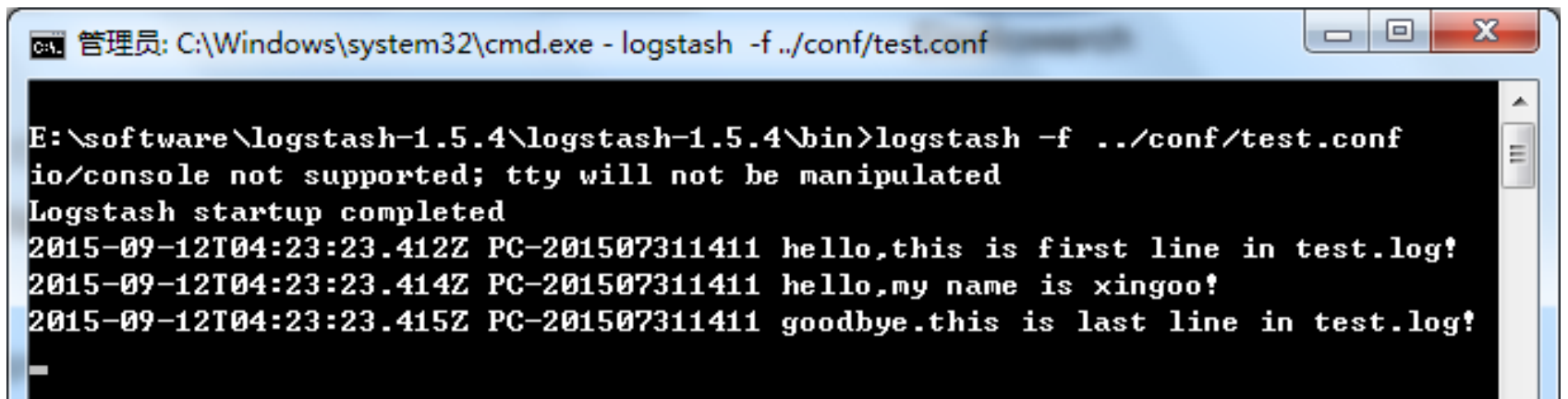
每个阶段都由很多的插件配合工作，比如file、elasticsearch、redis等等。

每个阶段也可以指定多种方式，比如输出既可以输出到elasticsearch中，也可以指定到stdout在控制台打印。

由于这种插件式的组织方式，使得logstash变得易于扩展和定制。

## 命令行中常用的命令

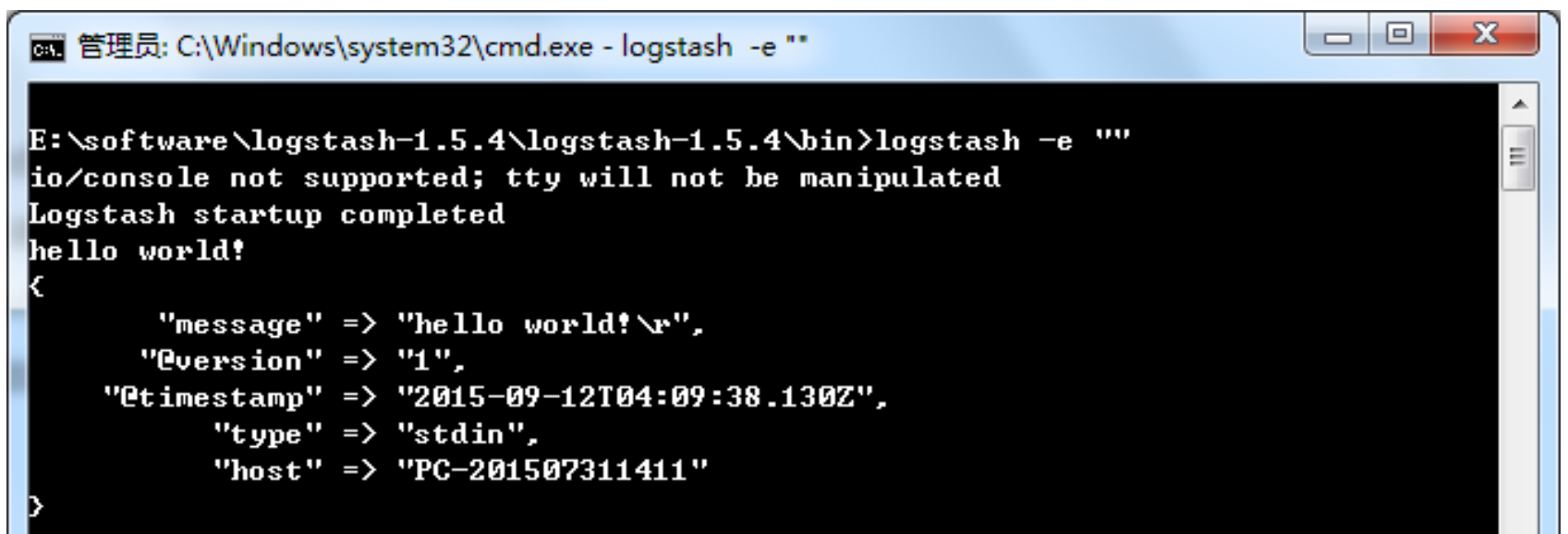
-f: 通过这个命令可以指定Logstash的配置文件，根据配置文件配置logstash



```
C:\Windows\system32\cmd.exe - logstash -f ../conf/test.conf

E:\software\logstash-1.5.4\logstash-1.5.4\bin>logstash -f ../conf/test.conf
io/console not supported; tty will not be manipulated
Logstash startup completed
2015-09-12T04:23:23.412Z PC-201507311411 hello,this is first line in test.log!
2015-09-12T04:23:23.414Z PC-201507311411 hello,my name is xingoo!
2015-09-12T04:23:23.415Z PC-201507311411 goodbye.this is last line in test.log!
```

-e: 后面跟着字符串，该字符串可以被当做logstash的配置（如果是“”则默认使用stdin作为输入， stdout作为输出）

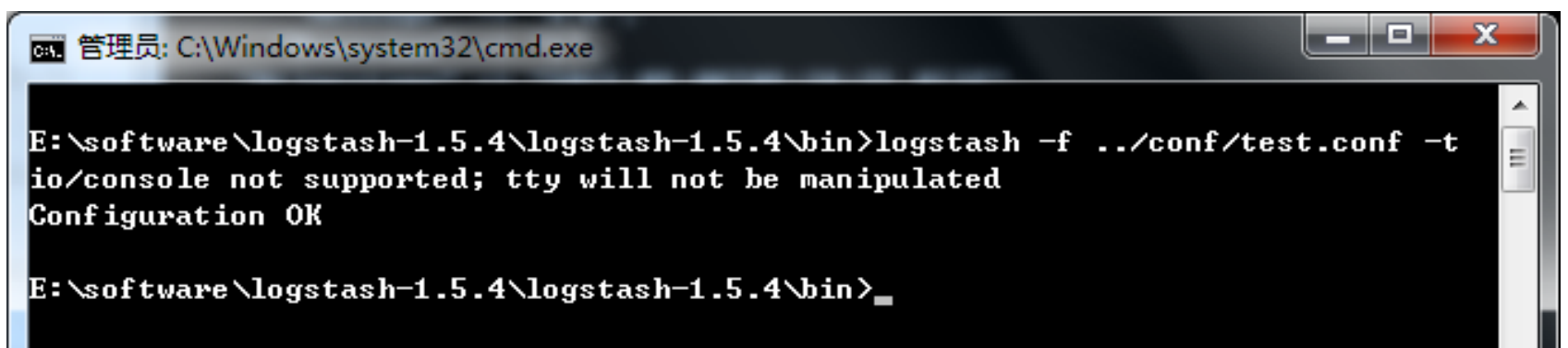


```
C:\Windows\system32\cmd.exe - logstash -e ""

E:\software\logstash-1.5.4\logstash-1.5.4\bin>logstash -e ""
io/console not supported; tty will not be manipulated
Logstash startup completed
hello world!
{
  "message" => "hello world!\r",
  "@version" => "1",
  "@timestamp" => "2015-09-12T04:09:38.130Z",
  "type" => "stdin",
  "host" => "PC-201507311411"
}
```

-l: 日志输出的地址（默认就是stdout直接在控制台中输出）

-t: 测试配置文件是否正确，然后退出。



```
C:\Windows\system32\cmd.exe

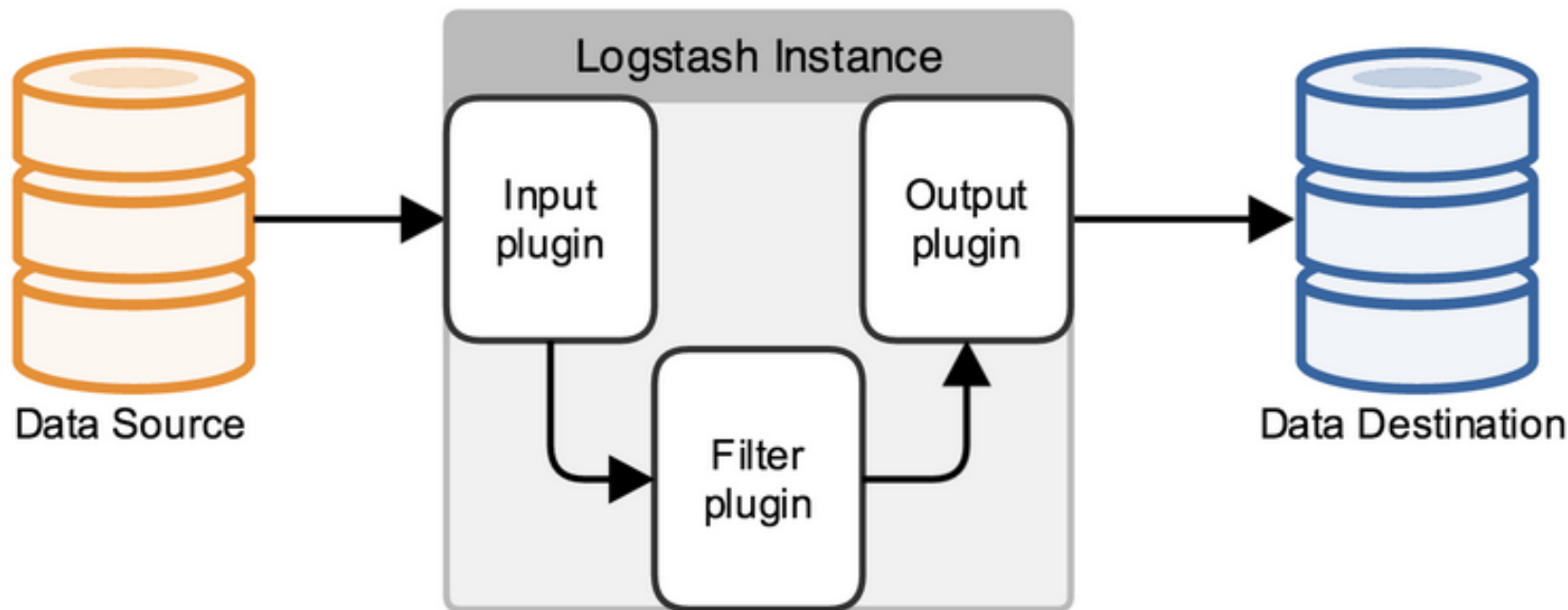
E:\software\logstash-1.5.4\logstash-1.5.4\bin>logstash -f ../conf/test.conf -t
io/console not supported; tty will not be manipulated
Configuration OK

E:\software\logstash-1.5.4\logstash-1.5.4\bin>
```

# 配置文件说明

前面介绍过logstash基本上由三部分组成，input、output以及用户需要才添加的filter，因此标准的配置文件格式如下：

```
input {...}
filter {...}
output {...}
```



在每个部分中，也可以指定多个访问方式，例如我想要指定两个日志来源文件，则可以这样写：

```
input {
  file { path => "/var/log/messages" type => "syslog" }
  file { path => "/var/log/apache/access.log" type => "apache" }
}
```

类似的，如果在filter中添加了多种处理规则，则按照它的顺序一一处理，但是有一些插件并不是线程安全的。

比如在filter中指定了两个一样的的插件，这两个任务并不能保证准确的按顺序执行，因此官方也推荐避免在filter中重复使用插件。

说完这些，简单的创建一个配置文件的小例子看看：

```
input {
  file {
    #指定监听的文件路径，注意必须是绝对路径
    path => "E:/software/logstash-1.5.4/logstash-1.5.4/data/test.log"
    start_position => beginning
  }
}
filter {

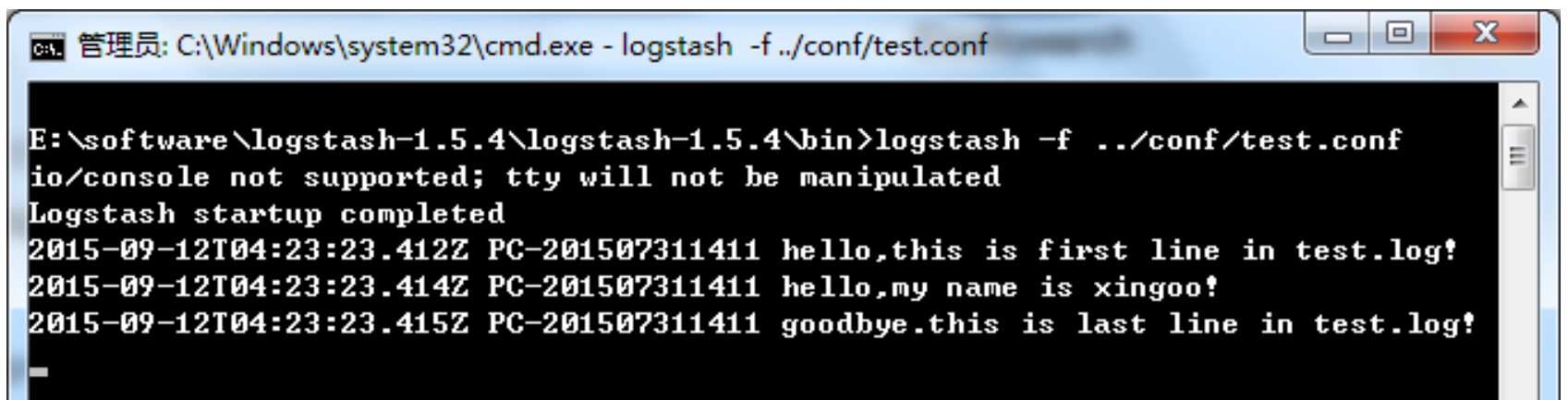
}
output {
  stdout {}
}
```

日志大致如下：

```
1 hello,this is first line in test.log!
2 hello,my name is xingoo!
3 goodbye.this is last line in test.log!
4
```

注意最后有一个空行。

执行命令得到如下信息：

A screenshot of a Windows command prompt window. The title bar reads "管理员: C:\Windows\system32\cmd.exe - logstash -f ../conf/test.conf". The command prompt shows the following output:

```
E:\software\logstash-1.5.4\logstash-1.5.4\bin>logstash -f ../conf/test.conf
io/console not supported; tty will not be manipulated
Logstash startup completed
2015-09-12T04:23:23.412Z PC-201507311411 hello,this is first line in test.log!
2015-09-12T04:23:23.414Z PC-201507311411 hello,my name is xingoo!
2015-09-12T04:23:23.415Z PC-201507311411 goodbye.this is last line in test.log!
```

细心的会发现，这个日志输出与上面的logstash -e "" 并不相同，这是因为上面的命令默认指定了返回的格式是json形式。

至此，就是logstash入门篇的介绍了，稍后会介绍关于logstash更多的内容，感兴趣的可以关注哦！

## 参考

【1】 Logstash官方文档：

<https://www.elastic.co/guide/en/logstash/current/index.html>