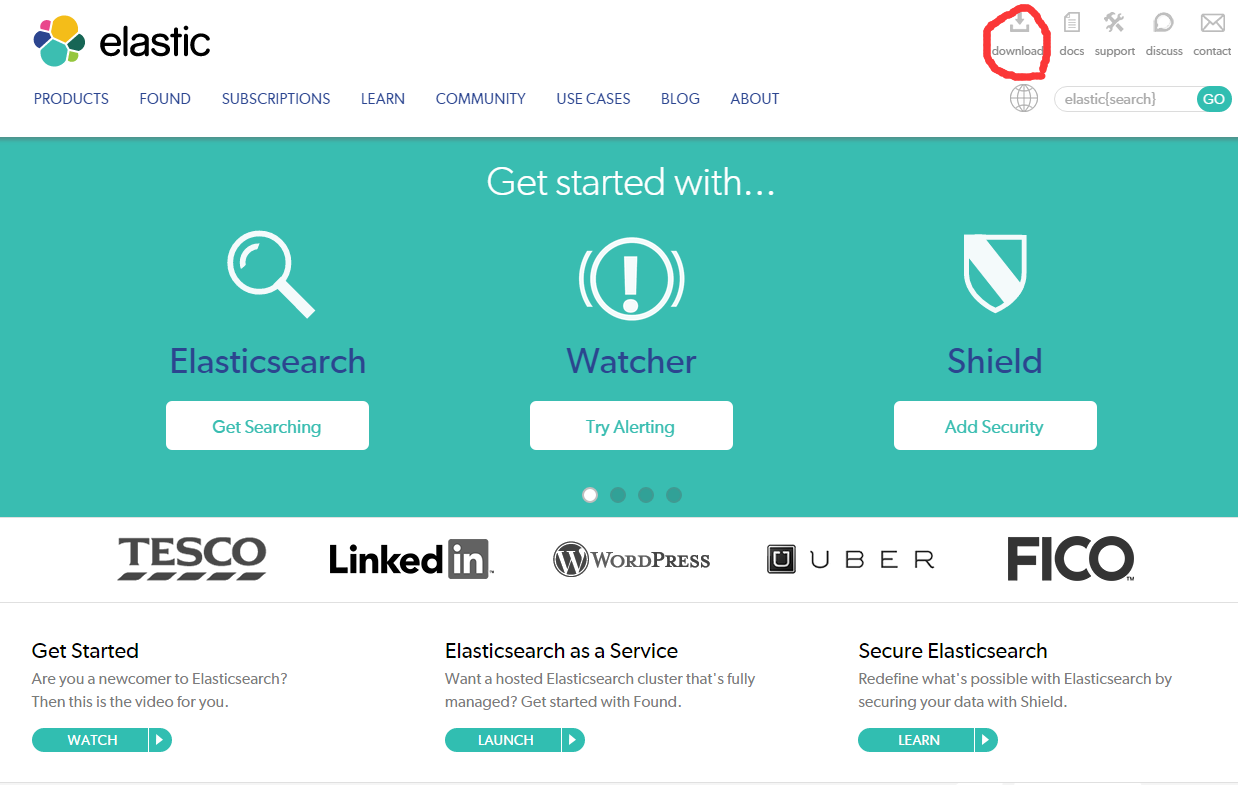
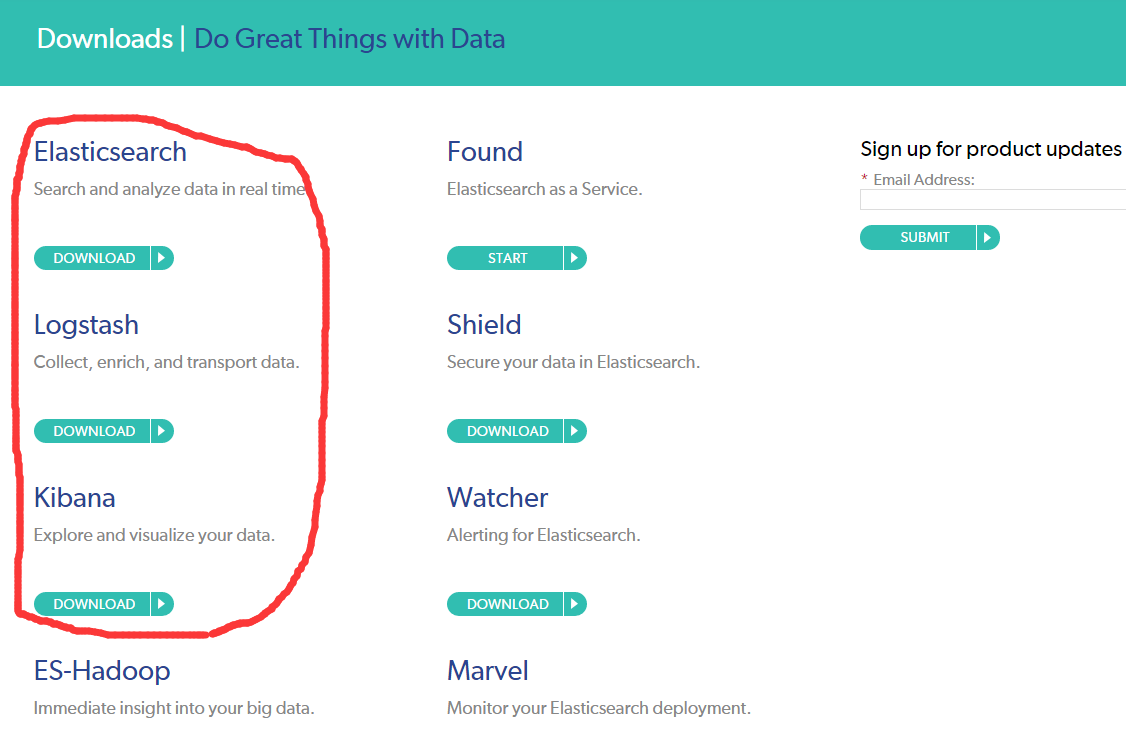
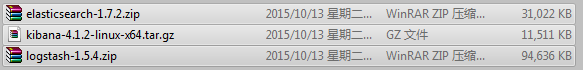
一：下载&解压

https://www.elastic.co

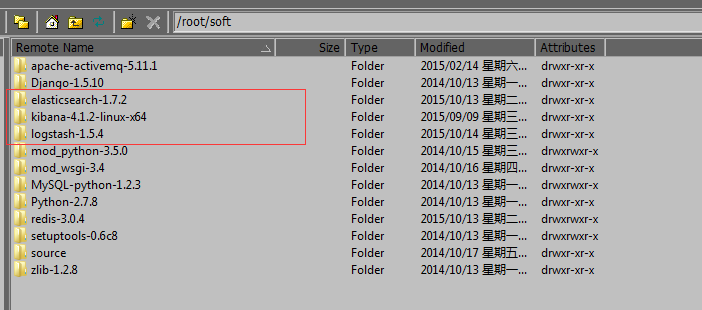




这里是我下载的版本：



解压到指定文件夹：



二：安装&启动logstash

解压后直接执行：

/root/soft/logstash-1.5.4/bin/logstash -e 'input { stdin {} } output { stdout {} }'

等待启动完成，控制台会输出如下日志：

Logstash shutdown completed

之后在控制台直接输入：hello world

logstash会自动输出hello world到控制台

如图：



启动logstash可配置参数有三个：input数据源、filter过滤器、output输出位置

input和output必选

上例中input为控制台输入，output为控制台输出

也可以将文件设置为数据源

input {

file {

type => "sidland"

path => "/usr/local/nginx/logs/www.sidland.net.access.log"

start\_position => beginning

}

}

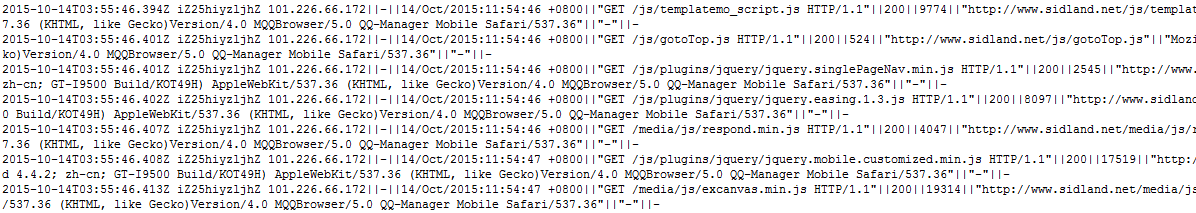
output {

stdout {}

}

三：启动测试

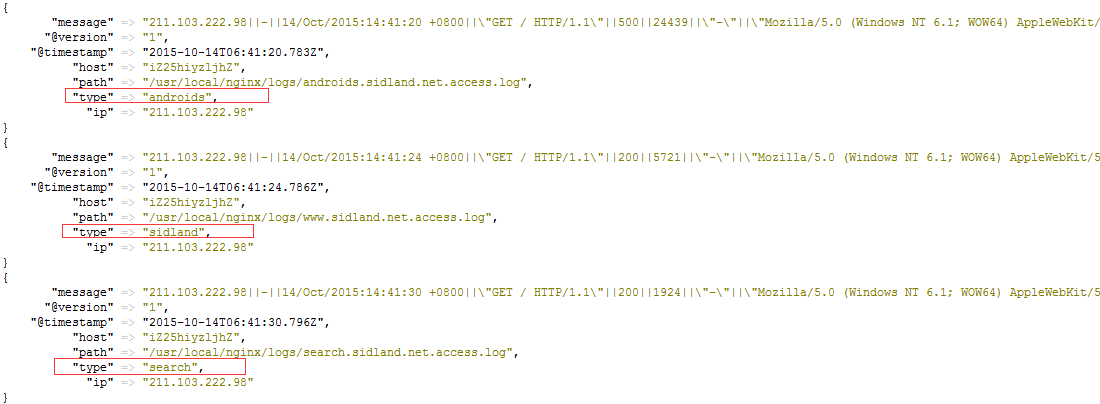
默认是原样输出：



可以做json格式：

在output中设置输出编码：

output {stdout { codec => rubydebug}}



四：启用过滤

input {

file {

type => "sidland"

path => "/usr/local/nginx/logs/www.sidland.net.access.log"

start\_position => beginning

}

file {

type => "androids"

path => "/usr/local/nginx/logs/androids.sidland.net.access.log"

start\_position => beginning

}

file {

type => "search"

path => "/usr/local/nginx/logs/search.sidland.net.access.log"

start\_position => beginning

}

}

filter {

if [type]=="sidland"{

mutate{

split => ["message","||"]

add\_field => { "clientIp" => "from %{[message][0]}" }

}

}

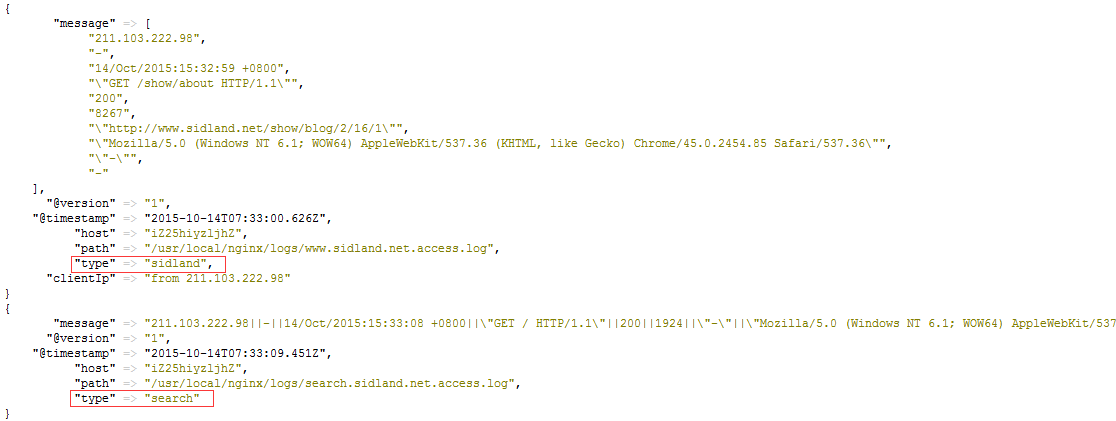
}

output {

stdout {codec => rubydebug}

}

测试输出结果：



从结果可以看出tpye类型为sidland的message进行了分割并增加了新的字段clientIp

而type类型为search的message并没有进行相应的处理

更多过滤属性详见官方文档

五：设置输出结果：

input {

file {

type => "sidland"

path => "/usr/local/nginx/logs/www.sidland.net.access.log"

start\_position => beginning

}

file {

type => "androids"

path => "/usr/local/nginx/logs/androids.sidland.net.access.log"

start\_position => beginning

}

file {

type => "search"

path => "/usr/local/nginx/logs/search.sidland.net.access.log"

start\_position => beginning

}

}

filter {

if [type]=="sidland"{

mutate{

split => ["message","||"]

add\_field => { "clientIp" => "from %{[message][0]}" }

}

}

}

output {

if [type]=="sidland"{

elasticsearch {

host => "127.0.0.1"

}

}

if [type]=="search"{

stdout {codec => rubydebug}

}

}

将type为sidland的日志输出到elasticsearch中

将type为search的日志输出到控制台中