ELK单机部署文档

```
author：apt
date：2019/01/14
usage：用于单机部署ELK日志分析系统的文档
```


一、ELK简介

什么是ELK？

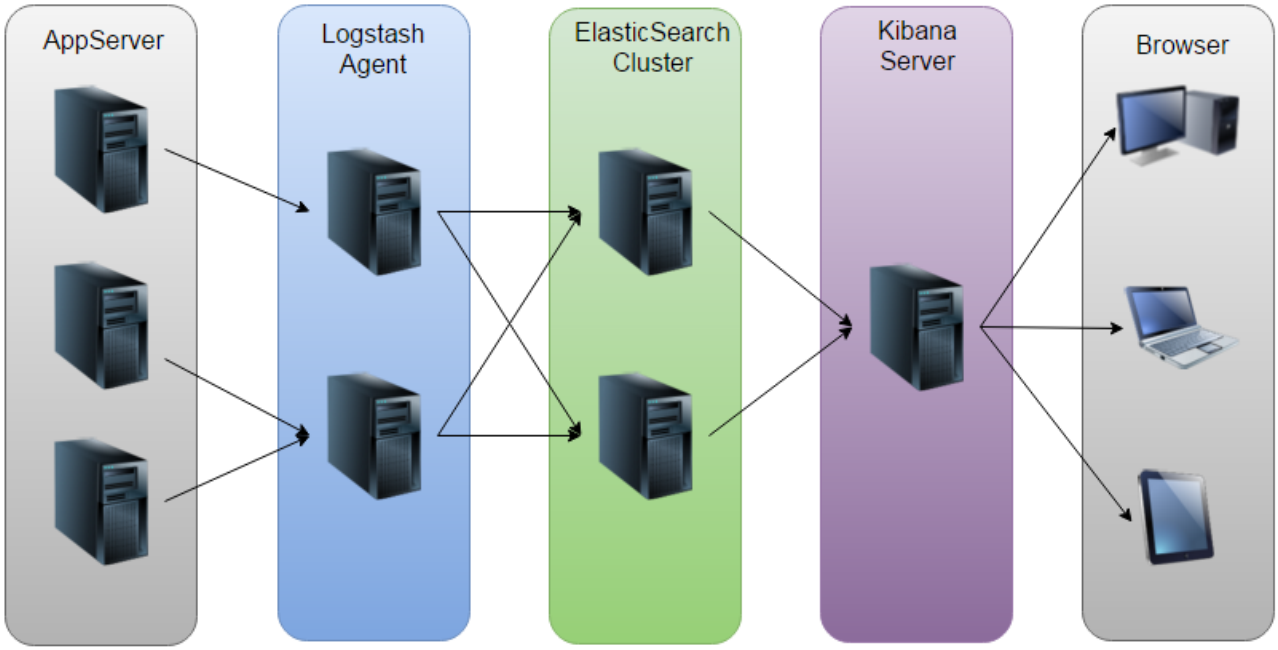
引用一段官方的话：

> what is the ELK？
>
> "ELK" is the acronym for three open source projects: Elasticsearch, Logstash, and Kibana. Elasticsearch is a search and analytics engine. Logstash is a server-side data processing pipeline that ingests data from multiple sources simultaneously, transforms it, and then sends it to a "stash" like Elasticsearch. Kibana lets users visualize data with charts and graphs in Elasticsearch.
>
> The Elastic Stack is the next evolution of the ELK Stack.

ELK主要是由3个开源项目组成的：elasticsearch，logstash，kibana

ELK工作原理介绍

如下图：



> 通过使用Logstash收集并处理AppServer产生的Log，并存放到ElasticSearch集群中，而Kibana则从ES集群中查询数据生成图表，再返回给Browser查看。

ELK组件介绍

Elasticsearch

Elasticsearch是个开源分布式搜索引擎，提供**搜集**、**分析**、**存储数据**三大功能。它的特点有：分布式，零配置，自动发现，索引自动分片，索引副本机制，restful风格接口，多数据源，自动搜索负载等。

我觉得他们的广告说得挺好的，`简单的事情就该简单做`。

Logstash

Logstash 主要是用来日志的搜集、分析、过滤日志的工具，支持大量的数据获取方式。一般工作方式为c/s架构，client端安装在需要收集日志的主机上，server端负责将收到的各节点日志进行过滤、修改等操作在一并发往elasticsearch上去。

Kibana

Kibana 也是一个开源和免费的工具，Kibana可以为 Logstash 和 ElasticSearch 提供的日志分析友好的 Web 界面，可以帮助汇总、分析和搜索重要数据日志。

beats

Beats 是轻量型采集器的平台，从边缘机器向 Logstash 和 Elasticsearch 发送数据。

环境版本介绍

Key	Value
操作系统	CentOS 7.6
硬件配置	2C 8G
JDK	jdk-8u201（1.8x）
elasticsearch	6.5.4
Logstash	6.5.4
kibana	6.5.4
filebeat	6.5.4
内网IP地址	10.0.0.179
外网IP地址	47.112.34.66

二、ELK日志分析系统的部署

单机部署

由于这里是先用于测试的ELK日志分析系统，所以最后决定只使用一台机器来做测试，也就是把ELK所有的组件集成安装到了这一台服务器上面。

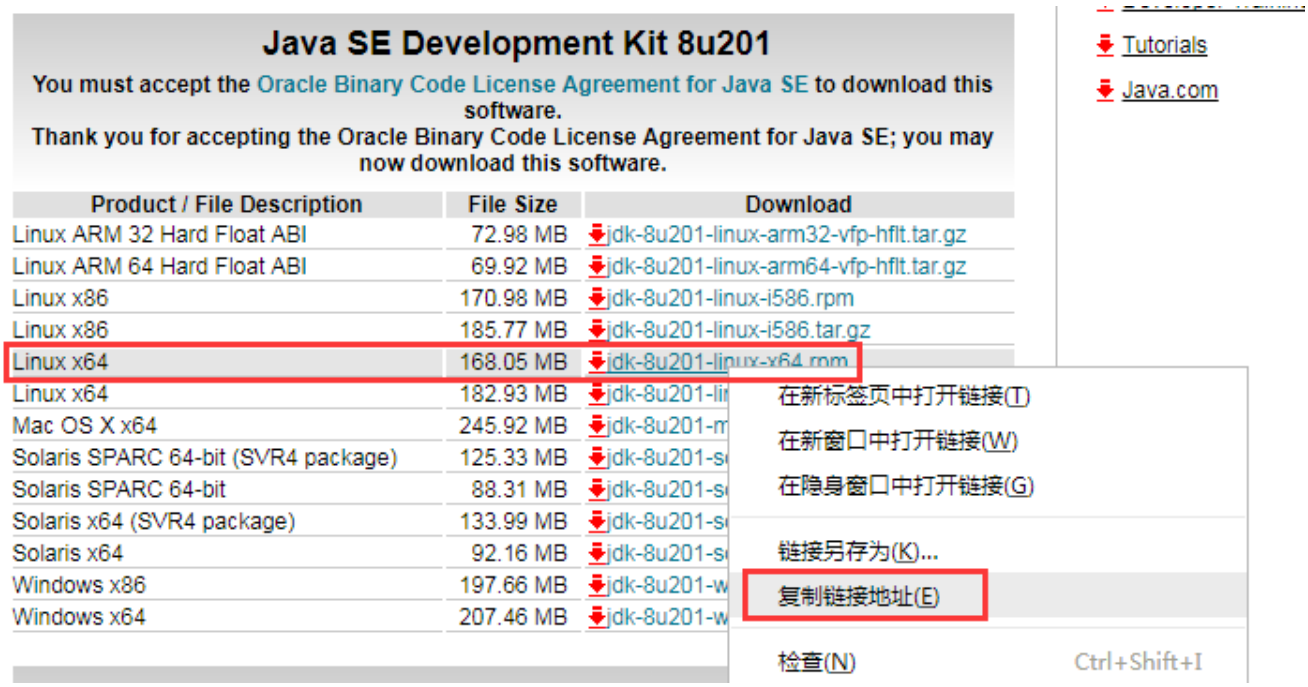
主机的优化/配置等

- 关闭防火墙
- 设置SELinux为disable等

安装JDK

- **下载官方的jdk rpm包进行安装**

1. 进入Oracle官网的https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html中找到合适的jdk的rpm包，然后点击右键后选择 `复制链接地址`



2. 然后在需要安装ELK的服务器中执行以下命令下载rpm包到服务器中：

```
# cd /usr/local/src
# wget --no-check-certificate --no-cookies --header "Cookie: oraclelicense=accept-securebackup-cookie" https://download.oracle.com/otn-pub/java/jdk/8u201-b09/42970487e3af4f5aa5bca3f542482c60/jdk-8u201-linux-x64.rpm
```

> 注：
>
> wget命令需要加入这个参数 --no-check-certificate --no-cookies --header "Cookie: oraclelicense=accept-securebackup-cookie" 否则会直接只下载一个网页文件，不会下载到真正想要的rpm包文件

3. 安装rpm包：

```
# yum -y localinstall jdk-8u201-linux-x64.rpm
```

4. 查看Java版本

```
# java -version
java version "1.8.0_201"
Java(TM) SE Runtime Environment (build 1.8.0_201-b09)
Java HotSpot(TM) 64-Bit Server VM (build 25.201-b09, mixed mode)
```

- **也可以使用yum源安装openjdk**

1. 检查yum源中是否含有1.8x版本的openjdk

```
# yum list | grep openjdk
```

2. 检查到有的话直接进行yum安装openjdk

```
# yum -y install java-1.8.0-openjdk
```

3. 检查Java版本

```
# java -version
openjdk version "1.8.0_191"
OpenJDK Runtime Environment (build 1.8.0_191-b12)
OpenJDK 64-Bit Server VM (build 25.191-b12, mixed mode)
```

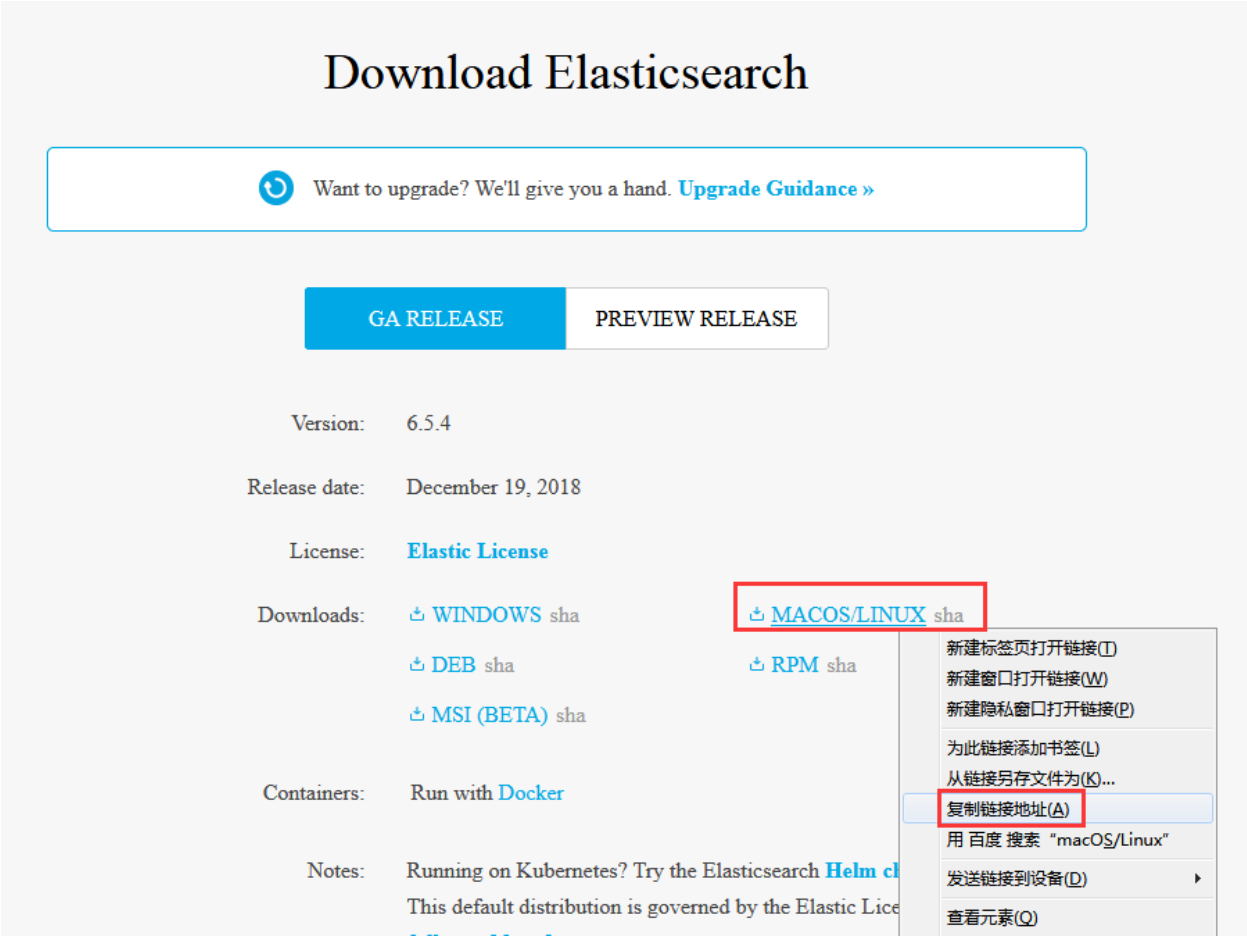
> 注：
>
> 由于openJDK去掉了JDK中涉及一些版权问题的API，所以会导致功能没有jdk那么完整，有可能会遇到一些问题，稳定性可能没那么好，所以并不推荐这个方式

安装Elasticsearch

这里测试就直接安装最新版本的elasticsearch，并且为了方便管理，采用的是源码安装的方式。

安装步骤如下：

1. 进入elasticsearch的下载页面，在浏览器地址栏中输入：`https://www.elastic.co/downloads/elasticsearch`，选择下载tar.gz包格式，鼠标右键复制连接地址。



2. 在需要安装elasticsearch的服务器中执行下列命令获取tar.gz包

```
# wget -O /usr/local/src/elasticsearch.tar.gz
https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-6.5.4.tar.gz
```

3. 创建一个安装elasticsearch的目录并把文件解压到指定这个指定的目录

```
# mkdir -p /usr/local/elasticsearch && tar xzvf /usr/local/src/elasticsearch.tar.gz
-C /usr/local/elasticsearch/ --strip-components 1
```

4. 备份配置文件elasticsearch.yml

```
# cp /usr/local/elasticsearch/config/elasticsearch.yml
/usr/local/elasticsearch/config/elasticsearch.yml.default
```

5. 修改配置文件elasticsearch.yml，修改后的文件内容如下

```
# vim /usr/local/elasticsearch/config/elasticsearch.yml
cluster.name: elk-application
node.name: node-1
path.data: /data/elasticsearch/data
path.logs: /data/elasticsearch/logs
network.host: 10.0.0.179
http.port: 9200
discovery.zen.ping.unicast.hosts: ["node-1"]
discovery.zen.minimum_master_nodes: 1
```

6. 创建elasticsearch用户，并给安装目录授权，因为不能直接用root用户启动

```
# useradd elasticsearch
# chown -R elasticsearch.elasticsearch /data/elasticsearch/
# chown -R elasticsearch.elasticsearch /usr/local/elasticsearch/
```

7. 修改sysctl.conf文件，添加一个vm.max_map_count参数（查了一下添加这个参数的数值是为了避免之后可能会报错说`max virtual memory areas vm.max_map_count [65530] is too low`）

```
# echo "vm.max_map_count = 262144" >> /etc/sysctl.conf
# sysctl -p
net.ipv6.conf.all.disable_ipv6 = 1
net.ipv6.conf.default.disable_ipv6 = 1
net.ipv6.conf.lo.disable_ipv6 = 1
vm.swappiness = 0
net.ipv4.neigh.default.gc_stale_time = 120
net.ipv4.conf.all.rp_filter = 0
net.ipv4.conf.default.rp_filter = 0
net.ipv4.conf.default.arp_announce = 2
net.ipv4.conf.lo.arp_announce = 2
net.ipv4.conf.all.arp_announce = 2
net.ipv4.tcp_max_tw_buckets = 5000
net.ipv4.tcp_syncookies = 1
net.ipv4.tcp_max_syn_backlog = 1024
net.ipv4.tcp_synack_retries = 2
kernel.sysrq = 1
vm.max_map_count = 262144
```

8. 修改 /etc/security/limits.conf 文件，修改打开文件句柄如下：

```
*               soft    nofile          100000
*               hard    nofile          100000
*               soft    nproc          100000
*               hard    nproc          100000
```

9. 添加信息ip和name到hosts文件，如下：

```
# echo "10.0.0.179 node-1" >> /etc/hosts
```

10. 启动elasticsearch

先切换到elasticsearch用户再进行启动，`nohup是一个比较简单的后台运行命令，之后会进行更多更高阶的启动使用`。

```
# su - elasticsearch
$ nohup /usr/local/elasticsearch/bin/elasticsearch &>/dev/null &
```

11. 检查验证elasticsearch的启动

    使用ps命令查看进程

```
$ ps aux | grep elasticsearch
root      3736  0.0  0.0 189712  2340 pts/3    S    15:02   0:00 su - elasticsearch
elastic+  4208  175 14.4 3625492 1158728 pts/1 Sl   15:08   0:03 /bin/java -Xms1g -
Xmx1g -XX:+UseConcMarkSweepGC -XX:CMSInitiatingOccupancyFraction=75 -
XX:+UseCMSInitiatingOccupancyOnly -XX:+AlwaysPreTouch -Xss1m -
Djava.awt.headless=true -Dfile.encoding=UTF-8 -Djna.nosys=true -XX:-
OmitStackTraceInFastThrow -Dio.netty.noUnsafe=true -
Dio.netty.noKeySetOptimization=true -Dio.netty.recycler.maxCapacityPerThread=0 -
Dlog4j.shutdownHookEnabled=false -Dlog4j2.disable.jmx=true -
Djava.io.tmpdir=/tmp/elasticsearch.UFomLl1j -XX:+HeapDumpOnOutOfMemoryError -
XX:HeapDumpPath=data -XX:ErrorFile=logs/hs_err_pid%p.log -XX:+PrintGCDetails -
XX:+PrintGCDateStamps -XX:+PrintTenuringDistribution -
XX:+PrintGCApplicationStoppedTime -Xloggc:logs/gc.log -XX:+UseGCLogFileRotation -
XX:NumberOfGCLogFiles=32 -XX:GCLogFileSize=64m -
Des.path.home=/usr/local/elasticsearch -
Des.path.conf=/usr/local/elasticsearch/config -Des.distribution.flavor=default -
Des.distribution.type=tar -cp /usr/local/elasticsearch/lib/*
org.elasticsearch.bootstrap.Elasticsearch
elastic+  4262  0.0  0.0  63940  5116 pts/1    Sl   15:08   0:00
```

12. 简单的curl测试

    正常运行的时候会显示下面这样的内容，否则会直接提示 Connection refused

```
# curl 10.0.0.179:9200
{
  "name" : "node-1",
  "cluster_name" : "elk-application",
  "cluster_uuid" : "ppy7POkXR8qhocSvAhkaog",
  "version" : {
    "number" : "6.5.4",
    "build_flavor" : "default",
    "build_type" : "tar",
    "build_hash" : "d2ef93d",
    "build_date" : "2018-12-17T21:17:40.758843Z",
    "build_snapshot" : false,
    "lucene_version" : "7.5.0",
    "minimum_wire_compatibility_version" : "5.6.0",
    "minimum_index_compatibility_version" : "5.0.0"
  },
  "tagline" : "You Know, for Search"
}
```

安装Logstash与filebeat

filebeat用于在各个服务器上获取数据，发送到logstash上，再由logstash处理数据。

1. 在官网的Logstash下载页面中，获取tar.gz包的下载链接
2. 在服务器中使用wget下载tar包

```
# wget -O /usr/local/src/logstash.tar.gz
https://artifacts.elastic.co/downloads/logstash/logstash-6.5.4.tar.gz
```

3. 解压

```
# mkdir -p /usr/local/logstash && tar xzvf /usr/local/src/logstash.tar.gz -C
/usr/local/logstash/ --strip-components 1
```

4. **安装filebeat**
    1. 在官网中获取filebeat的下载链接
    2. 在服务器中使用wget下载tar包

```
```
wget -O /usr/local/src/filebeat.tar.gz
https://artifacts.elastic.co/downloads/beats/filebeat/filebeat-6.5.4-linux-
x86_64.tar.gz
```
```

    1. 解压

```
   # mkdir -p /usr/local/filebeat && tar xzvf /usr/local/src/filebeat.tar.gz -C
/usr/local/filebeat/ --strip-components 1
```

4. 修改配置文件

```
   # cat filebeat.yml
   filebeat.prospectors:
   - input_type: log
     paths:
       - /var/log/message-log  # 测试本机的一个log文件
   output.logstash:
     hosts: ["10.0.0.179:5044"]
```

5. 启动filebeat服务

```
# /usr/local/filebeat/filebeat &
```

6. 检查启动，filebeat没有监听端口，主要看进程和日志

> filebeat监听的文件记录信息在/data/filebeat/data/registry

```
# ps aux | grep filebeat
root      8219  0.0  0.1 235616 15736 pts/1    Sl   16:23   0:00 ./filebeat
root      9648  0.0  0.0 112708   980 pts/2    S+   16:51   0:00 grep --color=auto
filebeat
# tail -f /usr/local/filebeat/logs/filebeat
2019-01-22T16:47:53.018+0800    INFO    [monitoring]    log/log.go:144  Non-zero
metrics in the last 30s        {"monitoring": {"metrics": {"beat":{"cpu":{"system":
{"ticks":90,"time":{"ms":1}},"total":{"ticks":210,"time":
{"ms":6},"value":210},"user":{"ticks":120,"time":{"ms":5}}},"handles":{"limit":
{"hard":65535,"soft":65535},"open":6},"info":{"ephemeral_id":"7b89263a-c980-47b1-
8cde-d53ebf96a61d","uptime":{"ms":1470015}},"memstats":
{"gc_next":4194304,"memory_alloc":1563304,"memory_total":16943072}},"filebeat":
{"harvester":{"open_files":0,"running":0}},"libbeat":{"config":{"module":
{"running":0}},"pipeline":{"clients":1,"events":{"active":0}}},"registrar":
{"states":{"current":0}},"system":{"load":{"1":0,"15":0.05,"5":0.01,"norm":
{"1":0,"15":0.025,"5":0.005}}}}}}
```

7. 新建一个本地文件message-log，可以取几条本机系统的messages文件，内容如下：

```
# cat message-log
Jan 20 03:00:01 apt systemd: Removed slice User Slice of root.
Jan 20 03:01:01 apt systemd: Created slice User Slice of root.
Jan 20 03:01:01 apt systemd: Started Session 91 of user root.
Jan 20 03:10:01 apt systemd: Started Session 92 of user root.
Jan 20 03:16:01 apt rsyslogd: [origin software="rsyslogd" swVersion="8.24.0-34.el7"
x-pid="3041" x-info="http://www.rsyslog.com"] rsyslogd was HUPed
```

8. 不指定文件启动Logstash测试

```
[root@apt bin]# ./logstash -e 'input { stdin {} } output {stdout {} }'
Sending Logstash logs to /usr/local/logstash/logs which is now configured via
log4j2.properties
[2019-01-23T17:20:14,772][WARN ][logstash.config.source.multilocal] Ignoring the
'pipelines.yml' file because modules or command line options are specified
[2019-01-23T17:20:14,795][INFO ][logstash.runner          ] Starting Logstash
{"logstash.version"=>"6.5.4"}
[2019-01-23T17:20:18,882][INFO ][logstash.pipeline        ] Starting pipeline
{:pipeline_id=>"main", "pipeline.workers"=>2, "pipeline.batch.size"=>125,
"pipeline.batch.delay"=>50}
[2019-01-23T17:20:19,074][INFO ][logstash.pipeline        ] Pipeline started
successfully {:pipeline_id=>"main", :thread=>"#<Thread:0x4784cb26 sleep>"}
The stdin plugin is now waiting for input:
[2019-01-23T17:20:19,151][INFO ][logstash.agent           ] Pipelines running
{:count=>1, :running_pipelines=>[:main], :non_running_pipelines=>[]}
[2019-01-23T17:20:19,550][INFO ][logstash.agent           ] Successfully started
Logstash API endpoint {:port=>9600}
hello world
{
       "message" => "hello world ",
          "host" => "node-1",
```

```
        "@version" => "1",
      "@timestamp" => 2019-01-23T09:20:26.883Z
  }
```

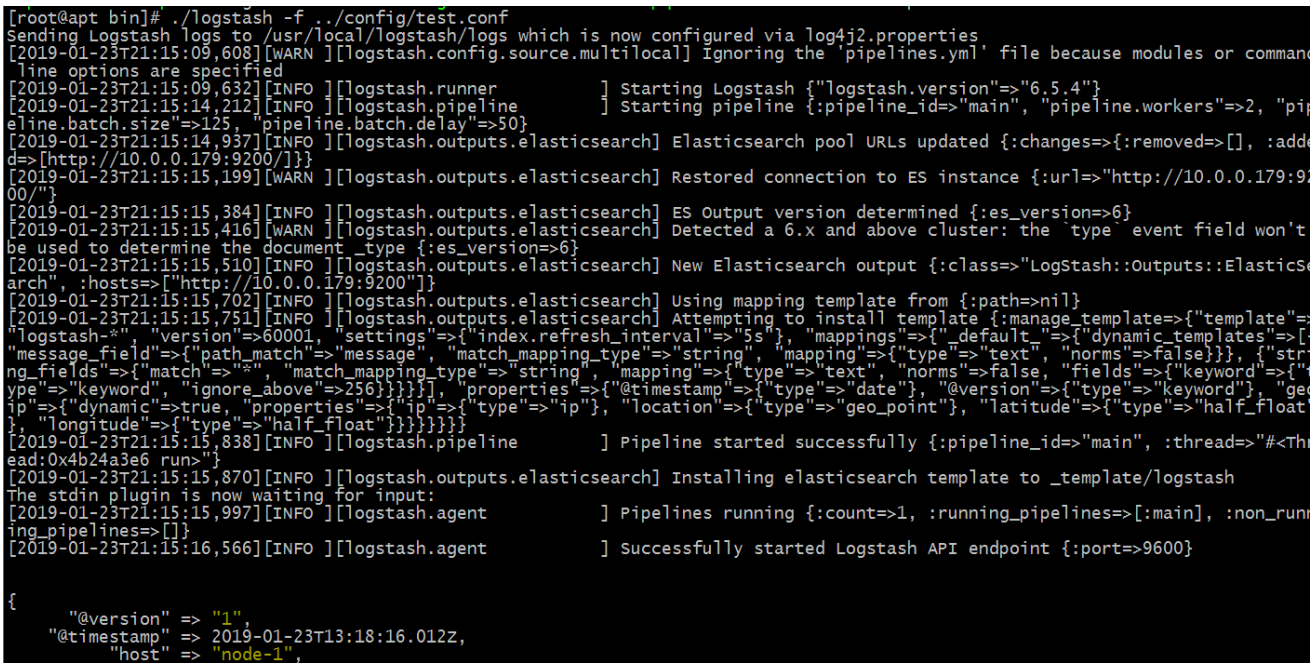
```
[root@apt bin]# ./logstash -e 'input { stdin {} } output {stdout {} }'
Sending Logstash logs to /usr/local/logstash/logs which is now configured via log4j2.properties
[2019-01-23T13:56:33,509][WARN ][logstash.config.source.multilocal] Ignoring the 'pipelines.yml' file because modules or command
 line options are specified
[2019-01-23T13:56:33,532][INFO ][logstash.runner          ] Starting Logstash {"logstash.version"=>"6.5.4"}
[2019-01-23T13:56:37,416][INFO ][logstash.pipeline        ] Starting pipeline {:pipeline_id=>"main", "pipeline.workers"=>2, "pip
eline.batch.size"=>125, "pipeline.batch.delay"=>50}
[2019-01-23T13:56:37,604][INFO ][logstash.pipeline        ] Pipeline started successfully {:pipeline_id=>"main", :thread=>"#<Thr
ead:0x6a54709c run>"}
The stdin plugin is now waiting for input:
[2019-01-23T13:56:37,689][INFO ][logstash.agent           ] Pipelines running {:count=>1, :running_pipelines=>[:main], :non_runn
ing_pipelines=>[]}
[2019-01-23T13:56:37,994][INFO ][logstash.agent           ] Successfully started Logstash API endpoint {:port=>9600}
hello world
{
          "host" => "node-1",
    "@timestamp" => 2019-01-23T05:56:45.959Z,
       "message" => "hello world",
      "@version" => "1"
}
```

9. 编写一个Logstash启动需要指定的配置文件

```
# cat /usr/local/logstash/config/test.conf
input {
  stdin {

  }
}

output {
  elasticsearch {
    hosts => "http://10.0.0.179:9200"
    index => "test-%{+YYYY.MM.dd}"
  }

 stdout {
  codec =>rubydebug
 }
}
```

10. 指定配置文件启动Logstash

```
# # cd /usr/local/logstash/
# ./logstash -f ../config/test.conf
```



11. 暂无

安装Kibana

1. 进入官网Kibana的下载页面，获取tar.gz包的下载连接
2. 在服务器中使用wget下载包

```
# wget -O /usr/local/src/kibana.tar.gz
https://artifacts.elastic.co/downloads/kibana/kibana-6.5.4-linux-x86_64.tar.gz
```

3. 解压

```
# mkdir -p /usr/local/kibana && tar xzvf /usr/local/src/kibana.tar.gz -C
/usr/local/kibana/ --strip-components 1
```

4. 修改配置文件（使用默认的端口即可，指定0.0.0.0为所有人都能访问，elasticsearch.url填写之前elasticsearch里面配置的url即可）

```
# cat /usr/local/kibana/config/kibana.yml
server.port: 5601
server.host: "0.0.0.0"
elasticsearch.url: "http://47.112.34.66:9200"
```

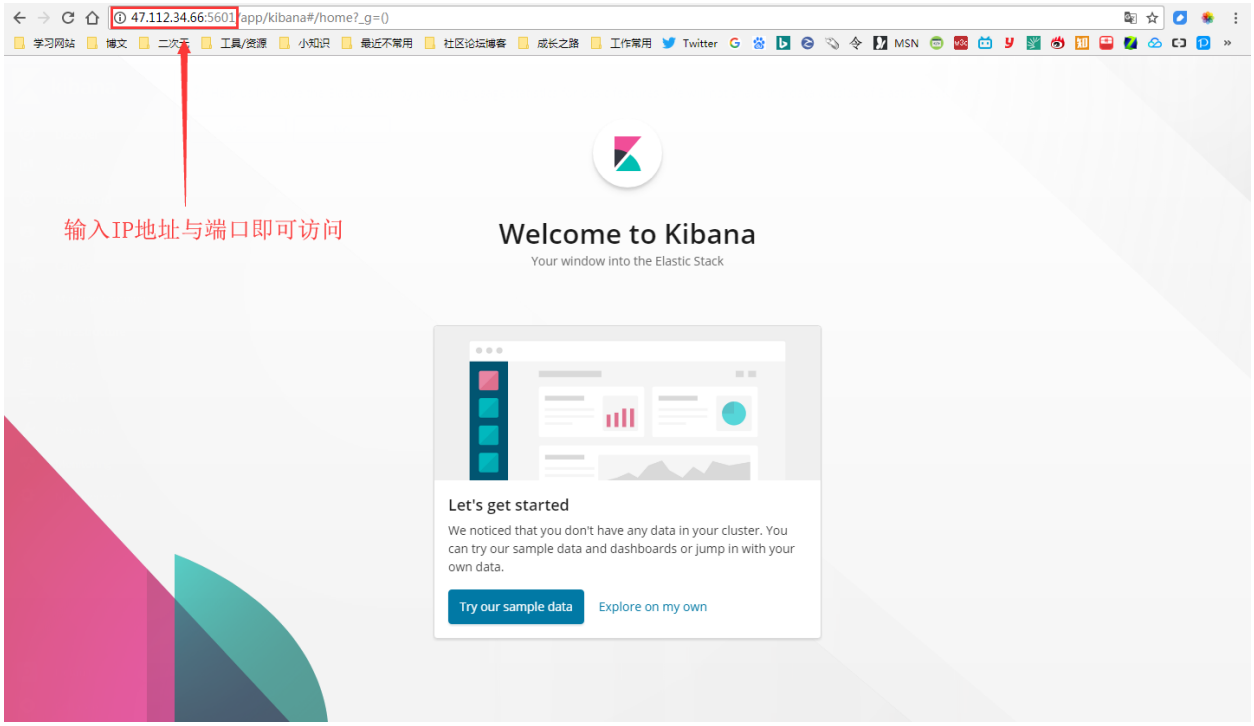
5. 启动进程

```
# nohup /usr/local/kibana/bin/kibana &>/dev/null  &
```

6. netstat检查一下启动状态

```
# netstat -ntlup | grep 5601
tcp        0      0 0.0.0.0:5601            0.0.0.0:*               LISTEN
 14677/node
```

7. 通过浏览器进行 ip:port 访问kibana的web界面



8. 如果需要停止kibana进程就执行下面的命令

```
# ps aux | grep -v grep | grep /node/bin/node | awk -F " " '{print $2}' | xargs
kill -9
```

9. 至于kibana的web页面和后续的ELK使用就留给之后的文档吧。