efk日志系统

安装文档

**日期：2018/07/19**

[1. 概述 1](#_Toc24850)

[1.1. 配置规划 1](#_Toc31486)

[1.2. 各软件版本 1](#_Toc15639)

[1.3. 日志处理流程 1](#_Toc2789)

[1.4. 各服务启动顺序 2](#_Toc23073)

[1.5. 架构 2](#_Toc9128)

[1.6. JDK准备 2](#_Toc15823)

[1.7. 新建efk文件夹 2](#_Toc30170)

[1.8. 下载软件 3](#_Toc14891)

[1.8.1. 进入下载地址 3](#_Toc26640)

[1.8.2. 点击download按钮 3](#_Toc27973)

[2. ElasticSearch安装 4](#_Toc29352)

[2.1. 解压安装包到/opt/efk 4](#_Toc7785)

[2.2. 在根路径下创建es用户组及用户 4](#_Toc13976)

[2.2.1. 创建用户组 5](#_Toc28198)

[2.2.2. 创建用户 5](#_Toc4059)

[2.2.3. 更改efk所属用户 5](#_Toc14220)

[2.3. 创建data数据目录和日志目录，使用es用户 5](#_Toc10304)

[2.4. 修改配置文件 6](#_Toc13374)

[2.5. 修改 Linux下/etc/security/limits.conf文件设置 8](#_Toc30111)

[2.6. 修改配置 Linux下/etc/security/limits.d/90-nproc.conf文件设置 9](#_Toc16987)

[2.7. 修改配置 Linux下/etc/sysctl.conf文件设置 9](#_Toc9919)

[2.8. 启动 9](#_Toc25148)

[3. Node.js安装 11](#_Toc23672)

[3.1. 下载Node.js 11](#_Toc29597)

[3.2. 解压Node.js 11](#_Toc14547)

[3.3. 修改环境变量 12](#_Toc13493)

[3.4. 测试 12](#_Toc23603)

[4. ElasticSearch-head插件安装 12](#_Toc23353)

[4.1. 采用nodejs安装 12](#_Toc26837)

[4.2. 下载ElasticSearch-head 12](#_Toc7787)

[4.3. 解压 13](#_Toc10013)

[4.4. 安装Node.js 13](#_Toc668)

[4.5. npm install -g grunt-cli 13](#_Toc31120)

[4.6. npm install 13](#_Toc31182)

[4.7. 更改配置信息 13](#_Toc1394)

[4.7.1. elasticsearch-5x下的 config/elasticsearch.yml 13](#_Toc19035)

[4.7.2. elasticsearch-head下Gruntfile.js 14](#_Toc23891)

[4.7.3. 修改连接地址 14](#_Toc29635)

[4.8. grunt server 15](#_Toc9496)

[4.9. 访问 15](#_Toc19225)

[5. Kibnan安装 15](#_Toc9029)

[5.1. 解压 15](#_Toc5029)

[5.2. 更改配置信息 15](#_Toc4493)

[5.3. 运行 16](#_Toc15788)

[5.4. 访问 16](#_Toc20217)

[5.5. 汉化 17](#_Toc1161)

[5.5.1. 下载汉化包 17](#_Toc26655)

[5.5.2. 解压汉化包 17](#_Toc11899)

[5.5.3. 运行main.pay文件 17](#_Toc20034)

[6. FileBeat安装 18](#_Toc967)

[6.1. 解压 18](#_Toc7259)

[6.2. 创建配置文件 18](#_Toc42)

[6.3. 启动 19](#_Toc329)

[7. Kafka安装 20](#_Toc6326)

[7.1. 下载 20](#_Toc24775)

[7.2. 解压 20](#_Toc378)

[7.3. 修改配置文件 20](#_Toc2268)

[7.4. 启动zookeeper 21](#_Toc31158)

[7.5. 启动kafka 22](#_Toc2004)

[7.6. 创建test主题 22](#_Toc6762)

[8. Logstash安装 22](#_Toc20233)

[8.1. 解压 22](#_Toc17012)

[8.2. 创建配置文件 23](#_Toc28942)

[8.3. 启动 25](#_Toc21667)

[9. 常见问题 25](#_Toc32236)

[9.1. ElasticSearch问题 25](#_Toc24425)

# 概述

## 配置规划

|  |  |  |
| --- | --- | --- |
| 机器名称 | 主从关系 | ip |
| cdh-1 | 主 | 10.168.1.44 |
| cdh-2 | 从 | 10.168.0.126 |
| cdh-3 | 从 | 10.168.0.127 |
| cdh-4 | 从 | 10.168.0.128 |
| cdh-5 | 从 | 10.168.0.130 |

## 各软件版本

|  |  |  |
| --- | --- | --- |
| 软件名称 | 版本号 | 插件 |
| Elasticsearch | 6.4.3 |  |
| Logstash | 6.4.3 |  |
| Kibana | 6.4.3 |  |
| Filebeat | 6.4.3 |  |
| Kafka | 2.11-1.1.0 |  |
| NodeJS | 8.11.3 |  |
| zookeeper | 3.4.5 |  |

## 日志处理流程

filebeat --> kafka --> logstash --> elasticsearch

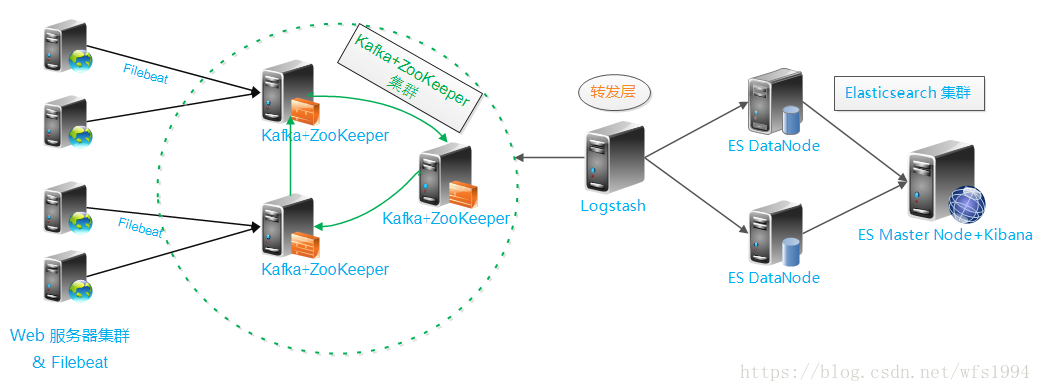
## 各服务启动顺序

启动顺序不能颠倒，filebeat一定放在最后启动。否则会出现logstash读取不到kafka数据的问题。

1 、elasticsearch、kafka、kibana

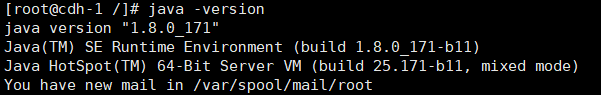
1. logstash
2. filebeat

## 架构



## JDK准备

5台机器都需安装1.8以上JDK，因搭建CDH集群时各机器已安装JDK1.8.0\_171，因此不再重复安装。



## 新建efk文件夹

5台机器都新建/opt/efk文件夹，efk相关软件都放到此文件夹下面。

|  |
| --- |
| mkdir efk |

## 下载软件

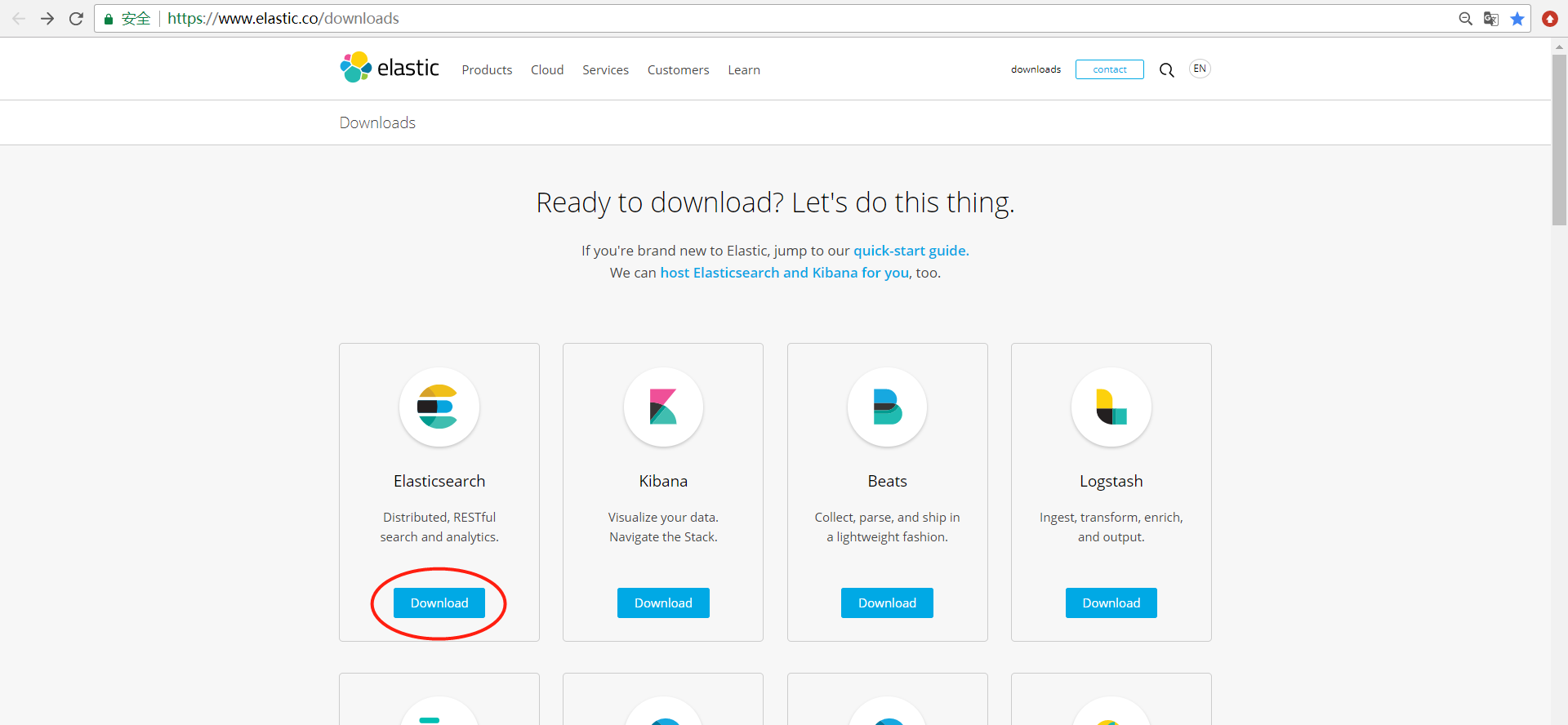
elastic系列软件下载地址：<https://www.elastic.co/downloads>

其中，Elasticsearch、Logstash、Kibana、Filebeat从地址下载。

此处以下载ElasticSearch为例，其他软件都参考步骤下载最新版本即可。

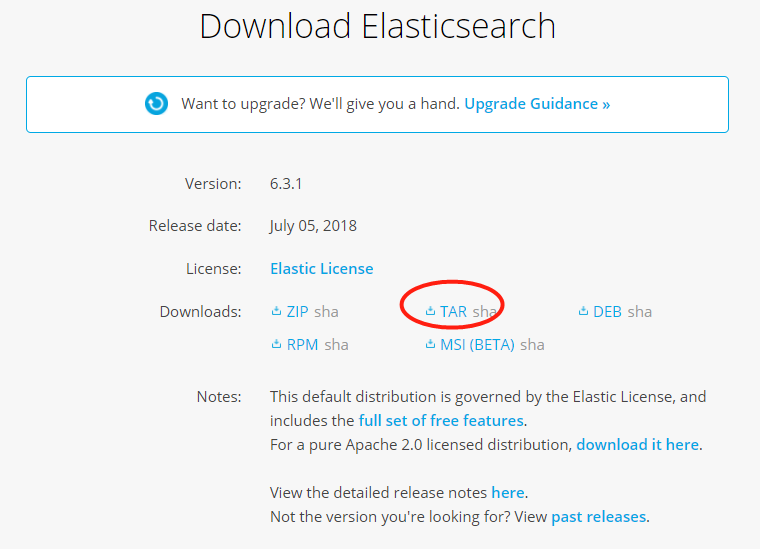
### 进入下载地址

浏览器输入<https://www.elastic.co/downloads>，进入下载地址



### 点击download按钮

点击download按钮，进入下载类型选择界面。点击下图中圈出的TAR选项，下载最新版本。本次安装下载的版本为elasticsearch-6.4.3.tar.gz



# ElasticSearch安装

## 解压安装包到/opt/efk

5台机器都需解压安装包。

包：elasticsearch-6.4.3.tar.gz

在/opt/efk解压：

|  |
| --- |
| tar -zxvf elasticsearch-6.4.3.tar.gz |

## 在根路径下创建es用户组及用户

回到/opt路径，5台机器依次创建相同的用户组及用户。

因elasticsearch版本在6.0之后不允许用root用户启动，故必须创建新的用户。

### 创建用户组

命令：

|  |
| --- |
| groupadd es |

### 创建用户

命令：

|  |
| --- |
| useradd es -g es -p elasticsearch |

### 更改efk所属用户

命令：

|  |
| --- |
| chown -R es:es efk |

## 创建data数据目录和日志目录，使用es用户

5台机器都在opt/efk路径下创建data目录和日志数据目录。

切换es用户：

|  |
| --- |
| su es |

创建data目录：

|  |
| --- |
| mkdir -p /opt/efk/elasticsearch\_data/data/ |

创建log目录：

|  |
| --- |
| mkdir -p /opt/efk/elasticsearch\_data/logs/ |

## 修改配置文件

cdh-1机器：vi opt/efk/elasticsearch-6.4.3/config/elasticsearch.yml

配置内容：

|  |
| --- |
| cluster.name: futuristEs  node.name: es-1  # 是否为master  node.master: true  # 是否为数据节点  node.data: true  # 数据目录  path.data: /opt/efk/elasticsearch\_data/data  # 日志目录  path.logs: /opt/efk/elasticsearch\_data/logs  bootstrap.memory\_lock: false  bootstrap.system\_call\_filter: false  # 本机IP  network.host: 0.0.0.0  # 本机http端口  http.port: 9200  # 指定集群中的节点中有几个有master资格的节点  discovery.zen.minimum\_master\_nodes: 1  # 指定集群中其他节点的IP  discovery.zen.ping.unicast.hosts: ["10.168.1.44","10.168.0.126","10.168.0.127","10.168.0.128","10.168.0.130"]  # ---------------------------------- 插件 -----------------------------------  # 下面两行配置为haad插件配置，5台服务器一致.  http.cors.enabled: true  http.cors.allow-origin: "\*" |

其余四台机器：vi opt/efk/elasticsearch-6.4.3/config/elasticsearch.yml

其中，node.name分别为：es-2，es-3，es-4，es-5。

master为false，其余都一样。

|  |
| --- |
| cluster.name: futuristEs  node.name: es-2  # 是否为master  node.master: false  # 是否为数据节点  node.data: true  # 数据目录  path.data: /opt/efk/elasticsearch\_data/data  # 日志目录  path.logs: /opt/efk/elasticsearch\_data/logs  bootstrap.memory\_lock: false  bootstrap.system\_call\_filter: false  # 本机IP  network.host: 0.0.0.0  # 本机http端口  http.port: 9200  # 指定集群中的节点中有几个有master资格的节点  discovery.zen.minimum\_master\_nodes: 1  # 指定集群中其他节点的IP  discovery.zen.ping.unicast.hosts: ["10.168.1.44","10.168.0.126","10.168.0.127","10.168.0.128","10.168.0.130"]  # ---------------------------------- 插件 -----------------------------------  # 下面两行配置为haad插件配置，5台服务器一致.  http.cors.enabled: true  http.cors.allow-origin: "\*" |

## 修改 Linux下/etc/security/limits.conf文件设置

5台机器都修改配置，最后配置如下

|  |
| --- |
| vi /etc/security/limits.conf  \* soft nofile 261444  \* hard nofile 262144  es soft memlock unlimited  es hard memlock unlimited |

## 修改配置 Linux下/etc/security/limits.d/90-nproc.conf文件设置

5台机器都修改配置，最后配置如下

|  |
| --- |
| vi /etc/security/limits.d/90-nproc.conf  \* soft nproc unlimited  root soft nproc unlimited |

## 修改配置 Linux下/etc/sysctl.conf文件设置

5台机器都修改配置，最后配置如下

|  |
| --- |
| vi /etc/sysctl.conf  vm.max\_map\_count = 262144  并执行命令：  sysctl -p |

## 启动

5台机器分布进入/opt/efk/elasticsearch-6.4.3/bin路径

|  |
| --- |
| ./elasticsearch |

后台长期启动

|  |
| --- |
| ./elasticsearch -d |

查找ES进程

|  |
| --- |
| ps -ef | grep elastic |

杀掉ES进程

|  |
| --- |
| kill -9 2382（进程号） |

重启ES

|  |
| --- |
| sh elasticsearch -d |

在浏览区输入：<http://10.168.0.127:9200/>

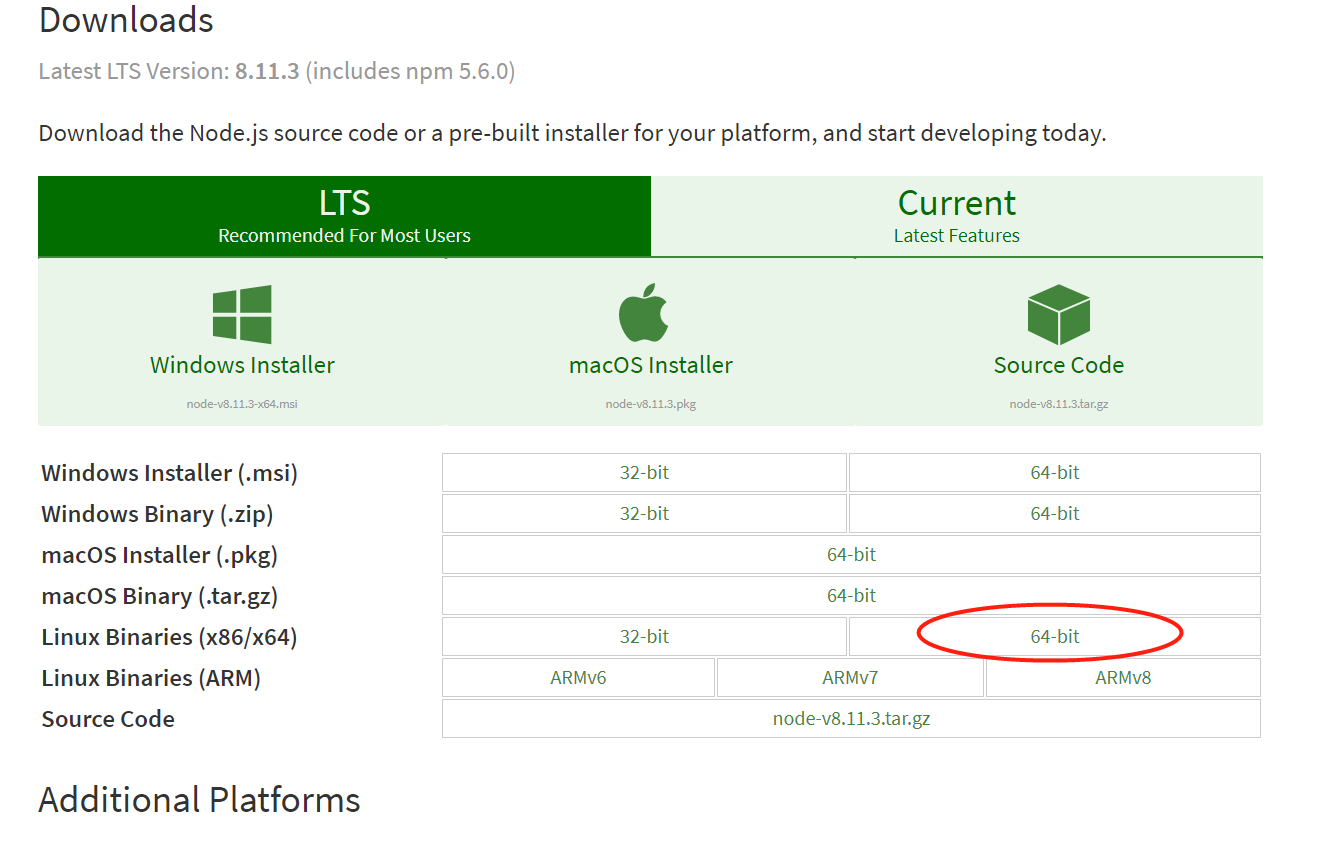
返回如下类似结果为正常启动

|  |
| --- |
| {  "name" : "es-3",  "cluster\_name" : "futuristEs",  "cluster\_uuid" : "\_na\_",  "version" : {  "number" : "6.4.3",  "build\_flavor" : "default",  "build\_type" : "tar",  "build\_hash" : "eb782d0",  "build\_date" : "2018-06-29T21:59:26.107521Z",  "build\_snapshot" : false,  "lucene\_version" : "7.3.1",  "minimum\_wire\_compatibility\_version" : "5.6.0",  "minimum\_index\_compatibility\_version" : "5.0.0"  },  "tagline" : "You Know, for Search"  } |

# Node.js安装

## 下载Node.js

打开官网下载链接:<https://nodejs.org/en/download/> ，下载的版本为：node-v8.11.3-linux-x64.tar.xz



## 解压Node.js

在cdh-1机器上解压node-v8.11.3-linux-x64.tar.xz。

opt/efk路径下：

|  |
| --- |
| tar -xvf node-v8.11.3-linux-x64.tar.xz |

## 修改环境变量

修改 /etc/profile

|  |
| --- |
| vi /etc/profile  export NODEJS\_HOME=/opt/efk/node-v8.11.3-linux-x64  export PATH=$PATH:$JAVA\_HOME/bin:$NODEJS\_HOME/bin |

执行source /etc/profile

|  |
| --- |
| source /etc/profile |

## 测试

出现v8.11.3说明测试成功



|  |
| --- |
| node -v  v8.11.3 |

# ElasticSearch-head插件安装

## 采用nodejs安装

6.X中，elasticsearch-head

不能放在elasticsearch的 plugins、modules 目录下

不能使用 elasticsearch-plugin install

## 下载ElasticSearch-head

下载地址<https://github.com/mobz/elasticsearch-head>

## 解压

在cdh-1机器上解压elasticsearch-head-master.zip。

opt/efk路径下：

|  |
| --- |
| unzip elasticsearch-head-master.zip |

## 安装Node.js

根据[Node.js安装](#_Node.js安装)，进行安装node.js

## npm install -g grunt-cli

/opt/efk/elasticsearch-head-master路径下：

|  |
| --- |
| npm install -g grunt-cli |

## npm install

/opt/efk/elasticsearch-head-master路径下：

|  |
| --- |
| npm install |

## 更改配置信息

### elasticsearch-5x下的 config/elasticsearch.yml

|  |
| --- |
| http.cors.enabled: true  http.cors.allow-origin: "\*" |

### elasticsearch-head下Gruntfile.js

|  |
| --- |
| connect: {  server: {  options: {  hostname: '0.0.0.0',  port: 9100,  base: '.',  keepalive: true  }  }  } |

### 修改连接地址

最后配置如下

|  |
| --- |
| vi /opt/efk/elasticsearch-head-master/\_site/app.js  找到如下代码  this.base\_uri = this.config.base\_uri || this.prefs.get("app-base\_uri") || "http://localhost:9200";  更改为  this.base\_uri = this.config.base\_uri || this.prefs.get("app-base\_uri") || "http://10.168.1.44:9200"; |

## grunt server

/opt/efk/elasticsearch-head-master路径下测试启动

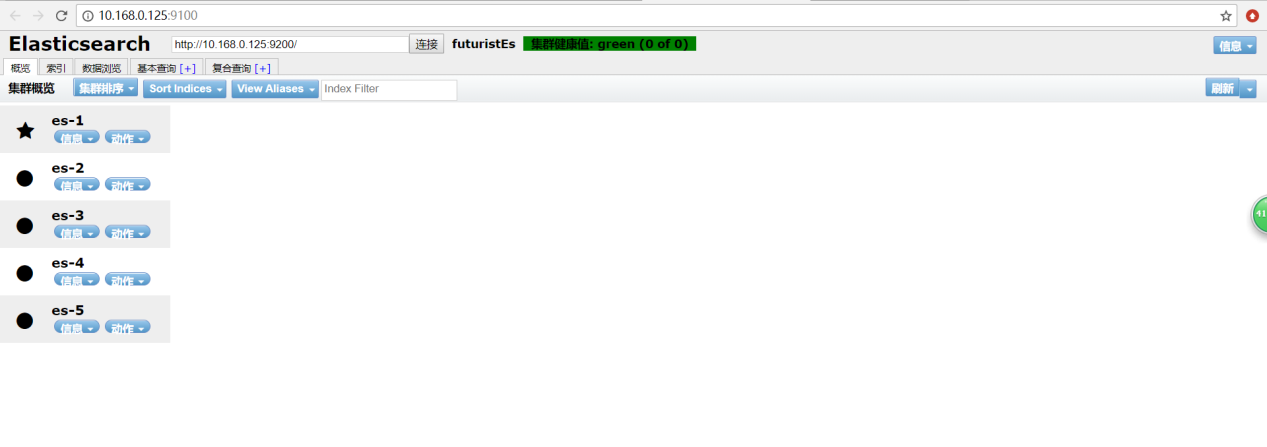
|  |
| --- |
| grunt server |

后台长期启动

|  |
| --- |
| nohup grunt server &exit |

## 访问

访问地址：[http://10.168.1.44:9100/](http://10.168.0.125:9100/)



# Kibnan安装

## 解压

注意版本要和es版本相同 。

在cdh-1机器上解压kibana-6.4.3-linux-x86\_64.tar.gz。

opt/efk路径下：

|  |
| --- |
| tar -zxvf kibana-6.4.3-linux-x86\_64.tar.gz |

## 更改配置信息

最后配置如下

|  |
| --- |
| vi /opt/efk/kibana-6.4.3-linux-x86\_64/config/kibana.yml  elasticsearch.url: "http://10.168.1.44:9200" # kibana监控哪台es机器  server.host: "10.168.1.44" # kibana运行在哪台机器 |

## 运行

Kibana6.3版本默认自带X-pack插件，故不需再次安装。

在/opt/efk/kibana-6.4.3-linux-x86\_64/bin路径下

|  |
| --- |
| ./kibana |

长期启动

|  |
| --- |
| nohup ./kibana &exit |

查找进程号

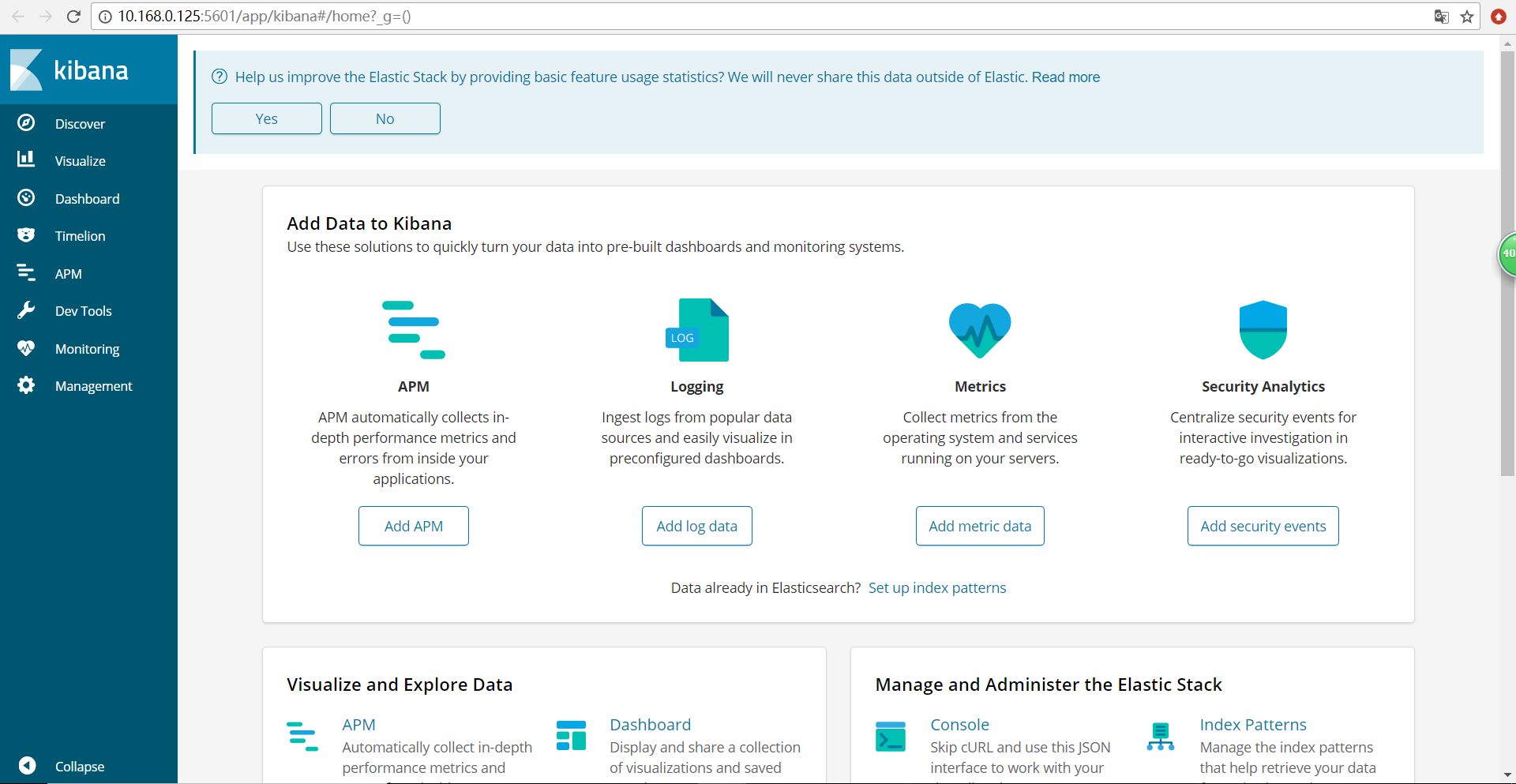
|  |
| --- |
| fuser -n tcp 5601 |

杀死进程号

|  |
| --- |
| kill -9 13265 |

## 访问

访问地址：http://10.168.1.44:5601



## 汉化

### 下载汉化包

github上有汉化的项目,地址:<https://github.com/anbai-inc/Kibana_Hanization>

### 解压汉化包

在/opt/efk/kibana-6.4.3-linux-x86\_64路径下

|  |
| --- |
| unzip Kibana\_Hanization-master.zip |

### 运行main.pay文件

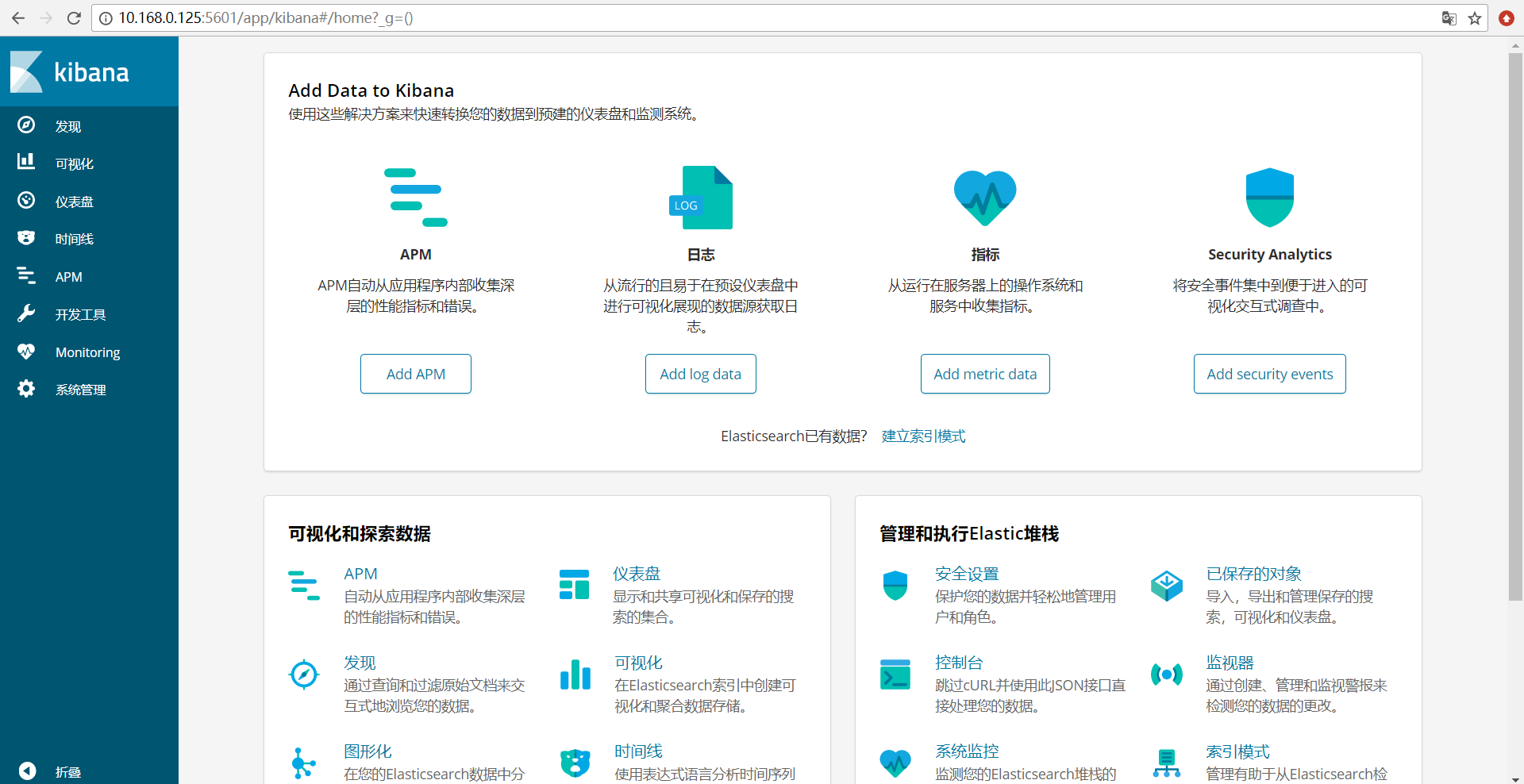
命令很简单,没错这是python脚本

在/opt/efk/kibana-6.4.3-linux-x86\_64/Kibana\_Hanization-master路径下

|  |
| --- |
| python main.py "/opt/efk/kibana-6.4.3-linux-x86\_64/" |

出现“恭喜，Kibana汉化完成！“表示成功

访问地址：http://10.168.1.44:5601



# FileBeat安装

## 解压

注意版本要和es版本相同 。

在5台机器上解压filebeat-6.4.3-linux-x86\_64.tar.gz。

opt/efk路径下：

|  |
| --- |
| tar -zxvf filebeat-6.4.3-linux-x86\_64.tar.gz |

## 创建配置文件

在/opt/efk/filebeat-6.4.3-linux-x86\_64路径下：

|  |
| --- |
| vi filebeat.yml |

配置内容为：(paths代表收集哪些文件夹下面的日志)

|  |
| --- |
| filebeat.inputs:  - type: log  enabled: false  paths:  - /var/log/audit/audit.log  filebeat.config.modules:  path: ${path.config}/modules.d/\*.yml  reload.enabled: false  setup.template.settings:  index.number\_of\_shards: 5  setup.kibana:  host: "10.168.1.44:5601"  output.kafka:  enabled: true  hosts: ["10.168.1.44:9092","10.168.0.126:9092","10.168.0.127:9092","10.168.0.128:9092","10.168.0.130:9092"]  topic: 'futurist' |

## 启动

/opt/efk/filebeat-6.4.3-linux-x86\_64路径下：

|  |
| --- |
| ./filebeat -e -c filebeat.yml -d "publish" |

长期启动：

|  |
| --- |
| nohup ./filebeat -e -c filebeat.yml &  centos7启动  sudo /bin/systemctl daemon-reload sudo /bin/systemctl enable filebeat.service |

查找进程ID并kill掉：

|  |
| --- |
| ps -ef |grep filebeat  kill -9 进程号 |

# Kafka（cdh集成）安装

## 下载

 csd包：<http://archive.cloudera.com/csds/kafka/>

这里下载的版本为：KAFKA-1.2.0.jar

 parcel包：<http://archive.cloudera.com/kafka/parcels/latest/>

这里下载的版本为：KAFKA-3.1.0-1.3.1.0.p0.35-el6.parcel和KAFKA-3.1.0-1.3.1.0.p0.35-el6.parcel.sha1

其中，KAFKA-3.1.0-1.3.1.0.p0.35-el6.parcel.sha1后缀需要改为.sha

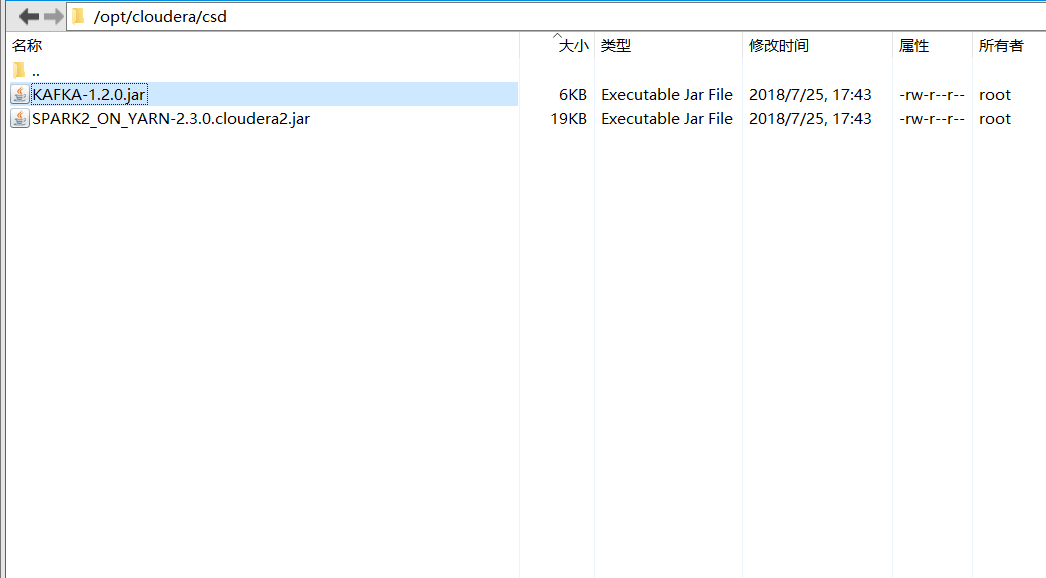
## 集成实现

### 关闭集群，关闭cm服务

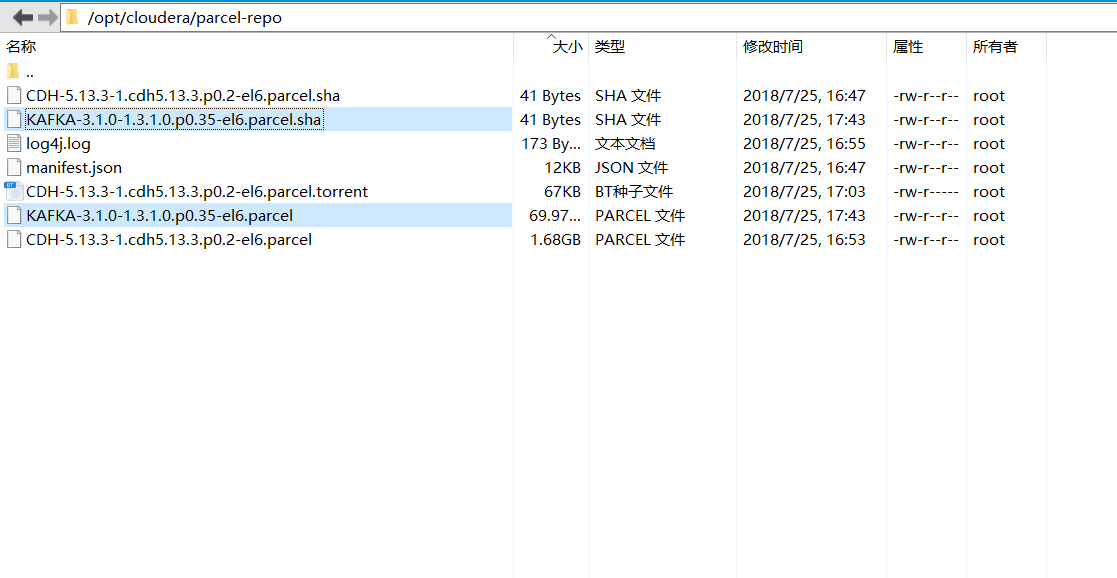
假如不关闭cm服务，会出现在添加kafka服务时找不到相关的服务描述。

### 上传包

将csd包放到cm安装节点下的 /opt/cloudera/csd目录下，如图 ：

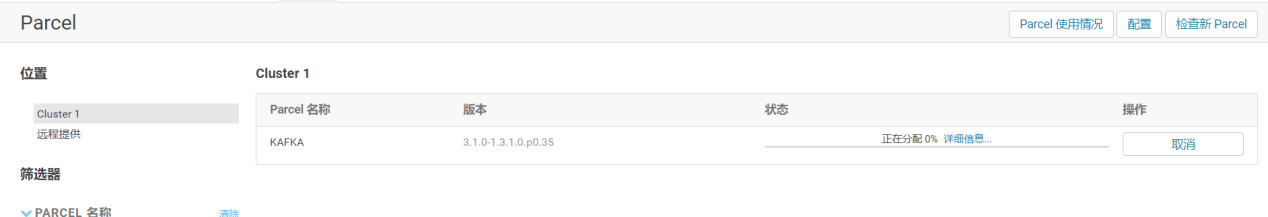


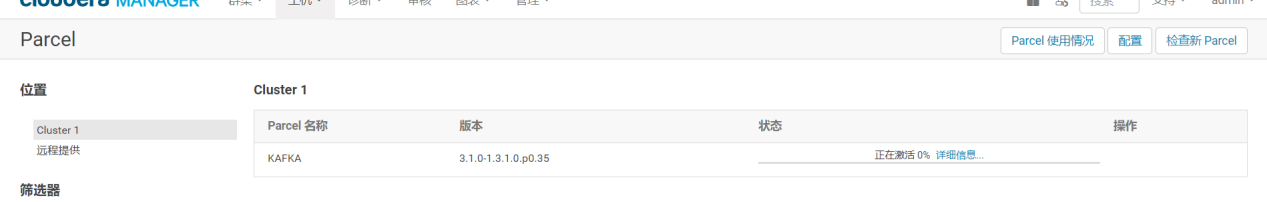
将parcel包放到cm安装节点下的 /opt/cloudera/parcel-repo目录下，如图：



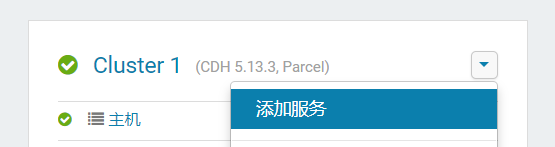
### 启动cm服务

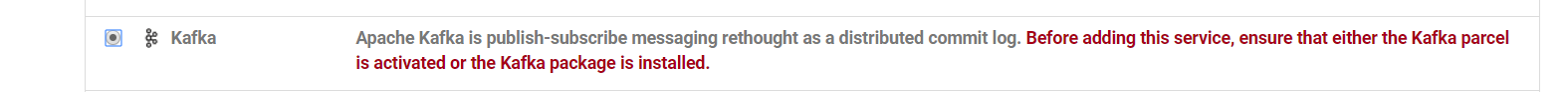
### 分配并激活percel包





### 添加kafka服务





其中，brokerId在此次安装中由于的对应关系为（由于之前本地安装过kafka，故brokerId需按下面配置，正常来说自动生成，不需配置）：

|  |  |
| --- | --- |
| 主机 | brokerId |
| cdh-1 | 1 |
| cdh-2 | 2 |
| cdh-3 | 5 |
| cdh-4 | 3 |
| cdh-5 | 4 |

### 启动kafka

# Kafka（本地）安装

## 下载

下载地址：<https://kafka.apache.org/downloads>

这里下载的版本为：kafka\_2.11-1.1.0.tgz

## 解压

在5台机器上解压kafka\_2.11-1.1.0.tgz。

opt/efk路径下：

|  |
| --- |
| tar -zxvf kafka\_2.11-1.1.0.tgz |

## 修改配置文件

cdh-1机器修改配置，最后配置如下

|  |
| --- |
| vi /opt/efk/kibana-6.4.3-linux-x86\_64/config  broker.id=1  listeners=PLAINTEXT://10.168.1.44:9092  zookeeper.connect=10.168.1.44:2181 |

cdh-2机器修改配置，最后配置如下

|  |
| --- |
| vi /opt/efk/kibana-6.4.3-linux-x86\_64/config  broker.id=2  listeners=PLAINTEXT://10.168.0.126:9092  zookeeper.connect=10.168.0.126:2181 |

cdh-3机器修改配置，最后配置如下

|  |
| --- |
| vi /opt/efk/kibana-6.4.3-linux-x86\_64/config  broker.id=3  listeners=PLAINTEXT://10.168.0.127:9092  zookeeper.connect=10.168.0.127:2181 |

cdh-4机器修改配置，最后配置如下

|  |
| --- |
| vi /opt/efk/kibana-6.4.3-linux-x86\_64/config  broker.id=4  listeners=PLAINTEXT://10.168.0.128:9092  zookeeper.connect=10.168.0.128:2181 |

cdh-5机器修改配置，最后配置如下

|  |
| --- |
| vi /opt/efk/kibana-6.4.3-linux-x86\_64/config  broker.id=5  listeners=PLAINTEXT://10.168.0.130:9092  zookeeper.connect=10.168.0.130:2181 |

## 启动zookeeper

$ bin/zookeeper-server-start.sh config/zookeeper.properties

默认使用的2181端口，可在配置文件修改。

## 启动kafka

/opt/efk/kafka\_2.11-1.1.0路径下：

|  |
| --- |
| ./bin/kafka-server-start.sh config/server.properties |

长期启动：

|  |
| --- |
| nohup ./bin/kafka-server-start.sh config/server.properties & |

查找进程号

|  |
| --- |
| fuser -n tcp 9092 |

杀死进程号

|  |
| --- |
| kill -9 13265 |

## 创建test主题

/opt/efk/kafka\_2.11-1.1.0路径下：

|  |
| --- |
| bin/kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 1 --partitions 5 --topic test  增加分区  bin/kafka-topics.sh --alter --zookeeper 10.168.0.126:2181 --topic futurist --partitions 5 |

# Logstash安装

## 解压

注意版本要和es版本相同 。

在cdh-1机器上解压logstash-6.4.3.tar.gz。

opt/efk路径下：

|  |
| --- |
| tar -zxvf logstash-6.4.3.tar.gz |

## 创建配置文件

在/opt/efk/logstash-6.4.3/config路径下：

|  |
| --- |
| vi kafka.conf |

配置内容为：

|  |
| --- |
| input {  kafka {  bootstrap\_servers => ["10.168.1.44:9092,10.168.0.126:9092,10.168.0.127:9092,10.168.0.128:9092,10.168.0.130:9092"]  group\_id =>"1"  topics => ["test"]  consumer\_threads => 5  decorate\_events =>true  codec => "json"  }  }  filter {  grok {match => { "message" => "%{DATA:timestamp}\|%{IP:serverIp}\|%{IP:clientIp}\|%{DATA:logSource}\|%{DATA:userId}\|%{DATA:reqUrl}\|%{DATA:reqUri}\|%{DATA:refer}\|%{DATA:device}\|%{DATA:textDuring}\|%{DATA:duringTime:int}\|\|"}}  }  output {  elasticsearch {  hosts => ["10.168.1.44:9200","10.168.0.126:9200","10.168.0.127:9200","10.168.0.128:9200","10.168.0.130:9200"]  index => "hdfs-%{type}-%{host}--%{+YYYY.MM.dd}"  codec => "json"  }  } |

## 启动

/opt/efk/logstash-6.4.3路径下：

|  |
| --- |
| ./bin/logstash -f config/kafka.conf |

长期启动：

|  |
| --- |
| nohup ./bin/logstash -f config/kafka.conf & |

查找ES进程

|  |
| --- |
| ps -ef | grep logstash |

杀掉ES进程

|  |
| --- |
| kill -9 2382（进程号） |

# 常见问题

## ElasticSearch问题

问题一：警告提示

[2016-11-06T16:27:21,712][WARN ][o.e.b.JNANatives ] unable to install syscall filter:

java.lang.UnsupportedOperationException: seccomp unavailable: requires kernel 3.5+ with CONFIG\_SECCOMP and CONFIG\_SECCOMP\_FILTER compiled in

at org.elasticsearch.bootstrap.Seccomp.linuxImpl(Seccomp.java:349) ~[elasticsearch-5.0.0.jar:5.0.0]

at org.elasticsearch.bootstrap.Seccomp.init(Seccomp.java:630) ~[elasticsearch-5.0.0.jar:5.0.0]

报了一大串错误，其实只是一个警告。

解决：使用新得linux版本，就不会出现此类问题了。

问题二：ERROR: bootstrap checks failed

max file descriptors [4096] for elasticsearch process likely too low, increase to at least [65536]

max number of threads [1024] for user [lishang] likely too low, increase to at least [2048]

解决：切换到root用户，编辑limits.conf 添加类似如下内容

vi /etc/security/limits.conf

添加如下内容:

\* soft nofile 65536

\* hard nofile 131072

\* soft nproc 2048

\* hard nproc 4096

问题三：max number of threads [1024] for user [lish] likely too low, increase to at least [2048]

解决：切换到root用户，进入limits.d目录下修改配置文件。

vi /etc/security/limits.d/90-nproc.conf

修改如下内容：

\* soft nproc 1024

#修改为

\* soft nproc 2048

问题四：max virtual memory areas vm.max\_map\_count [65530] likely too low, increase to at least [262144]

解决：切换到root用户修改配置sysctl.conf

vi /etc/sysctl.conf

添加下面配置：

vm.max\_map\_count=262144

并执行命令：

sysctl -p

问题五：max file descriptors [4096] for elasticsearch process likely too low, increase to at least [65536]

解决：修改切换到root用户修改配置limits.conf 添加下面两行

命令:vi /etc/security/limits.conf

\* hard nofile 65536

\* soft nofile 65536

切换到es的用户。

问题六：

ERROR: bootstrap checks failed

system call filters failed to install; check the logs and fix your configuration or disable system call filters at your own risk

原因：

这是在因为Centos6不支持SecComp，而ES5.2.0默认bootstrap.system\_call\_filter为true进行检测，所以导致检测失败，失败后直接导致ES不能启动。

解决：

在elasticsearch.yml中配置bootstrap.system\_call\_filter为false，注意要在Memory下面:

bootstrap.memory\_lock: false

bootstrap.system\_call\_filter: false

然后，重新启动elasticsearch，即可启动成功。

问题六：后台运行

最后还有一个小问题，如果你在服务器上安装Elasticsearch，而你想在本地机器上进行开发，这时候，你很可能需要在关闭终端的时候，让Elasticsearch继续保持运行。最简单的方法就是使用nohup。先按Ctrl + C，停止当前运行的Elasticsearch，改用下面的命令运行Elasticsearch

nohup./bin/elasticsearch&

这样，你就可以放心地关闭服务器终端，而不用担心Elasticsearch也跟着关闭了