# ELK抓取Tomcat日志

# 背景

通常，日志被分散的储存不同的设备上。如果你管理数十上百台服务器，你还在使用依次登录每台机器的传统方法查阅日志。这样是不是感觉很繁琐和效率低下。当务之急我们使用集中化的日志管理，例如：开源的 syslog ，将所有服务器上的日志收集汇总。

集中化管理日志后，日志的统计和检索又成为一件比较麻烦的事情，一般我们使用grep 、 awk 和 wc 等 Linux 命令能实现检索和统计，但是对于要求更高的查询、排序和统计等要求和庞大的机器数量依然使用这样的方法难免有点力不从心。

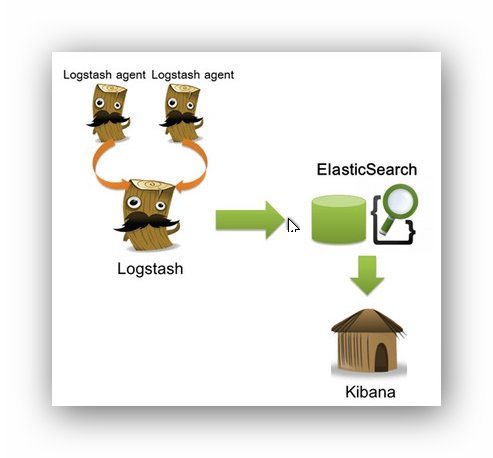
开源实时日志分析 ELK 平台能够完美的解决我们上述的问题， ELK 由 ElasticSearch 、 Logstash 和 Kiabana 三个开源工具组成。官方网站： [***https://www.elastic.co/products***](https://www.elastic.co/products)

l   Elasticsearch 是个开源分布式搜索引擎，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。

l   Logstash 是一个完全开源的工具，他可以对你的日志进行收集、分析，并将其存储供以后使用（如，搜索）。

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

工作原理如下如所示：



# 软件准备

## 从以下站点下载相关软件

Elasticsearch下载地址： <https://www.elastic.co/downloads/elasticsearch>

Logstash下载地址： <https://www.elastic.co/downloads/logstash>

Kibana下载地址： <https://www.elastic.co/downloads/kibana>

## 安装ELK相关软件

将下载的tar.gz文件上传至/opt/common文件夹中

分别执行tar -zxvf elasticsearch-2.3.5.tar.gz

tar -zxvf kibana-4.5.4-linux-x64.tar.gz

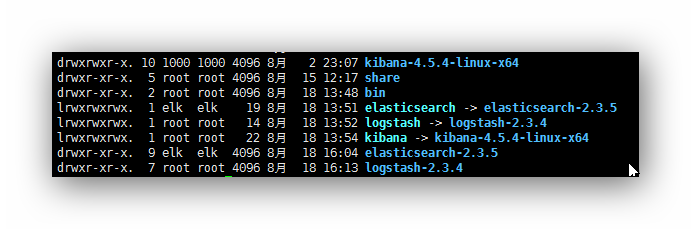
tar -zxvf logstash-2.3.4.tar.gz

cp -a elasticsearch-2.3.5 /usr/local

cd /usr/local

ln -s elasticsearch-2.3.5 elasticsearch 建立软连接

其他两个类似如上的操作，以上操作完成后在/usr/local目录下执行ls -trl命令会如下图



## 安装jdk

安装JDK8

上传jdk-8u45-linux-x64.tar.gz至/opt/common

tar -zxvf jdk-8u45-linux-x64.tar.gz

vim /etc/profile

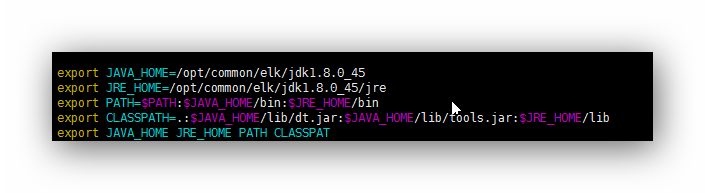
export JAVA\_HOME=/opt/common/elk/jdk1.8.0\_45

export JRE\_HOME=/opt/common/elk/jdk1.8.0\_45/jre

export PATH=$PATH:$JAVA\_HOME/bin:$JRE\_HOME/bin

export CLASSPATH=.:$JAVA\_HOME/lib/dt.jar:$JAVA\_HOME/lib/tools.jar:$JRE\_HOME/lib

export JAVA\_HOME JRE\_HOME PATH CLASSPAT



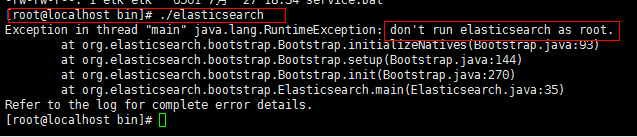
## 安装redis

安装并启动redis

使用ps –ef|grep redis 查看redis是否启动成功

# 运行elasticsearch

由于elasticsearch最新版要求不能使用root用户来启动程序，



## 所以需要新建用户

adduser elk 新增用户elk

passwd elk 为elk用户创建密码

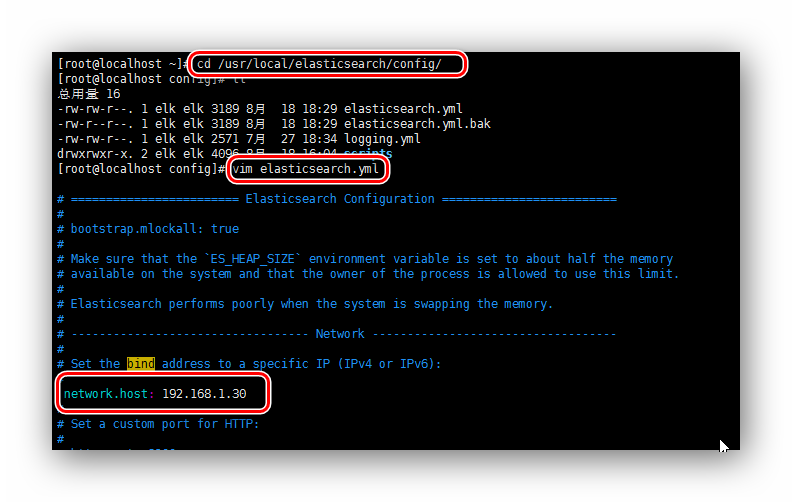
## 切换root账号su 回车输入密码

进入/usr/local/目录给elasticsearch进行赋权

chmod -R elk:elk

然后切换回elk用户 su elk

## 使用root修改es本地默认ip



## 在elasticsearch/bin目录下执行启动程序

/usr/local/elasticsearch/bin/elasticsearch -d

# 运行logstash

## 编辑配置信息

cd /usr/local/logstash

mkdir etc

vim central.conf

|  |
| --- |
| input {  redis {  host => "127.0.0.1"  port => "6379"  type => "redis-input"  data\_type => "list"  key => "logstash"  }  }  output {  elasticsearch {  hosts => ["192.168.1.30:9200"]  index => "logstash-%{type}-%{+YYYY.MM.dd}"  document\_type => "%{type}"  workers => 1  flush\_size => 20000  idle\_flush\_time => 10  template\_overwrite => true  }  } |

Redis 服务器是 logstash 官方推荐的 broker 选择。Broker 角色也就意味着会同时存在输入和输出俩个插件

参考：<http://udn.yyuap.com/doc/logstash-best-practice-cn/input/redis.html>

## 启动agent

/usr/local/logstash/bin/logstash agent --config /usr/local/logstash/etc/central.conf &

# 启动kibana

## cd /usr/local/kibana/config目录下

## vim kibana.yml修改

## elasticsearch.url: <http://192.168.1.30:9200>

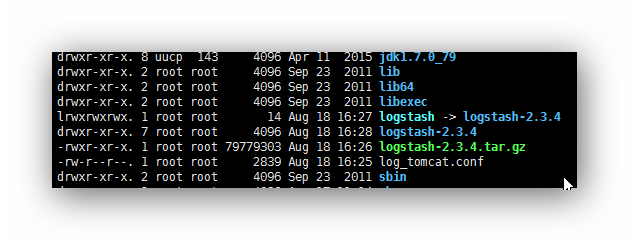
## cd /usr/local/kibana/bin

## ./kibana &

# 收集日志信息

## 进入待收集日志的系统我们已pms-web为例

## 上传logstash.gz到192.168.1.245并解压到/usr/local/目录中



## 建立软连接 ln –s logstash-2.3.4 logstash

## cd logstash 然后新建文件夹etc mkdir etc

## 进入etc目录vim log\_tomcat.conf

|  |
| --- |
| input {  file {  path => ["/usr/local/tomcat/logs/catalina.out","/usr/local/tomcat/logs/localhost.\*.log","/usr/local/tomcat/logs/localhost.\*.log"]  codec=> multiline {  pattern => "(^.+[^\[INFO\]]Exception:.+)|(^.+\[ERROR\].+)|(^[a-zA-Z])|(^\s+at .+)|(^\s+... \d+ more)|(^\s\*Caused by:.+)"  #pattern => "(^[a-zA-Z].+)|(^\s+at .+)|(^\s+... \d+ more)|(^\s\*Caused by:.+)"  #匹配这种日志，有的以[开头，有的直接以数字开头  #[04-21 15:42:00,123][DefaultQuartzScheduler\_Worker-6][INFO] carsing.crm.customer.service.impl.ServiceNoteWsServiceImpl.queryPeriodFromContract(line:742) CRM<<<<<<<<<Contract:<resultset></resultset>  #2016-04-21 15:42:15,022 [com.trade.info.impl.InfoPlatformDispatcherImpl:41]-[INFO] ---------线程开始提交--------  #pattern => "(^\s+)|(^=)|(^\d+=\d+)|(^\()|(^[a-zA-Z].+)|(^\s+at .+)|(^\s+... \d+ more)|(^\s\*Caused by:.+)"  #pattern => "(^[^\[])|(^\s+at .+)|(^\s\*Caused by:.+)"  #匹配这种日志，直接以[开头，即不是以[开头的都并入下一行  #[04-21 16:07:55,150][http-bio-8080-exec-837][INFO] carsing.crm.log.InfoInteractionWS.sysCarInfoToWechatService(line:304) >>>>>>>>>sysCarInfoToWechatService:  #pattern => "(^.+Exception:.+)|(^[a-zA-Z])|(^\s+at .+)|(^\s\*Caused by:.+)"  what=> "previous" #如果不换行的话放在哪，这里表示放在前面  }  }  }  filter {  if "ERROR" in [message] { #如果消息里有ERROR字符则将type改为自定义的标记  mutate { replace => { type => "tomcat\_catalina\_error" } }  }  else if "WARN" in [message] {  mutate { replace => { type => "tomcat\_catalina\_warn" } }  }  else {  mutate { replace => { type => "tomcat\_catalina\_info" } }  }  grok {  #match => { "message" => "%{COMBINEDAPACHELOG}" }  #match => [ "message", "%{TOMCATLOG}", "message", "%{CATALINALOG}" ]  match => [ "message", "\[%{MONTHNUM}-%{MONTHDAY}[T ]%{HOUR}:%{MINUTE}:(%{SECOND})\]\[(?<thread\_name>.+?)\]\[(?<log\_level>\w+)\]\s\*(?<content>.\*)", "message", "%{TIMESTAMP\_ISO8601:date} \[(?<thread\_name>.+?)\]-\[(?<log\_level>\w+)\]\s\*(?<content>.\*)" ]  #多种格式匹配，如下  #[04-21 15:42:00,123][DefaultQuartzScheduler\_Worker-6][INFO] carsing.crm.customer.service.impl.ServiceNoteWsServiceImpl.queryPeriodFromContract(line:742) CRM<<<<<<<<<Contract:<resultset></resultset>  #2016-04-21 15:42:15,022 [com.trade.info.impl.InfoPlatformDispatcherImpl:41]-[INFO] ---------线程开始提交--------  remove\_field => ["message"] #这表示匹配成功后是否删除原始信息，这个看个人情况，如果为了节省空间可以考虑删除  }  }  output {  redis {  host => "192.168.1.30"  port => "6379"  data\_type => "list"  key => "logstash"  }  } |

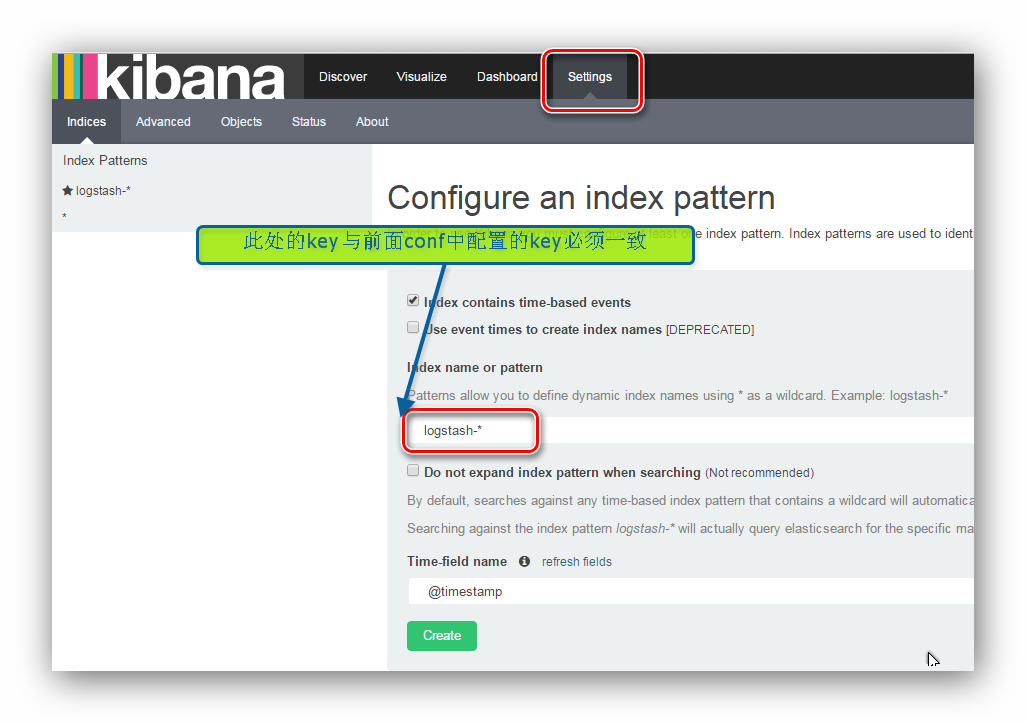
注意此处的key必须和服务端的key中的值一致

## 启动客户端进行日志收集

/usr/local/logstash/bin/logstash -f /usr/local/logstash/etc/log\_tomcat.conf &

# Kibana日志搜索

## 打开kibana主页<http://192.168.1.30:5601/>



配置完毕后点击create

## 选择Discover标签则可查看我们从192.168.1.245中tomcat里收集上来的日志

