**ELK入门及如何搭建**

1. **ELK是什么？**

在项目初期的时候，大家都是赶着上线，一般来说对日志没有过多的考虑，当然日志量也不大，所以用log4j就够了，随着应用的越来越多，日志散落在各个服务器的logs文件夹下，确实有点不大方便。

当我们需要日志分析的时候你大概会这么做：直接在日志文件中 grep、awk 就可以获得自己想要的信息。

那你们想过这种方式的问题吗？

1.日志量太大如何归档、文本搜索太慢怎么办、如何多维度查询

2.应用太多，面临数十上百台应用时你该怎么办

3.随意登录服务器查询log对系统的稳定性及安全性肯定有影响

4.如果使用人员对Linux不太熟练那面对庞大的日志简直要命

ELK因此就应运而生，那么为什么要用ELK呢？ELK又能给我们解决哪些问题呢？

1.日志统一收集，管理，访问。查找问题方便安全

2.使用简单，可以大大提高定位问题的效率

3.可以对收集起来的log进行分析

4.能够提供错误报告，监控机制

1. **ELK架构设计**

ELK是三个开源软件的缩写，分别表示：Elasticsearch , Logstash, Kibana , 它们都是开源软件

**1. LogStash**

     它可以流放到各自的服务器上收集Log日志，通过内置的ElasticSearch插件解析后输出到ES中

**2.ElasticSearch**

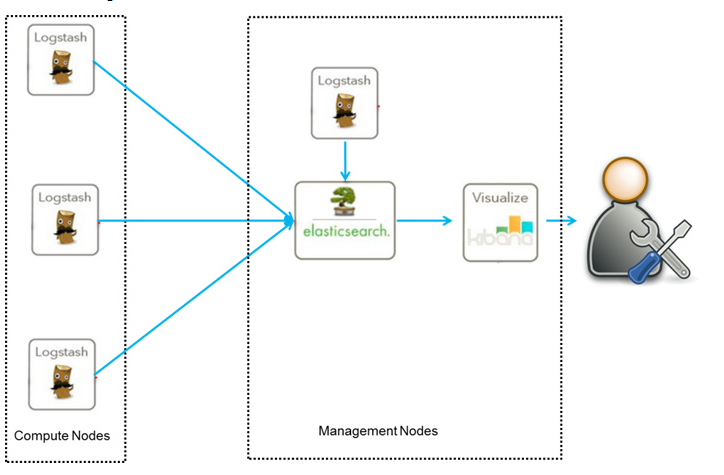
   这是一个基于Lucene的分布式全文搜索框架，可以对logs进行分布式存储，有点像hdfs。此为ELK的核心组件，日志的分析以及存储全部由es完成，因此在后面的课程中我们还会对此进行重点讲解。

**3. Kibana**

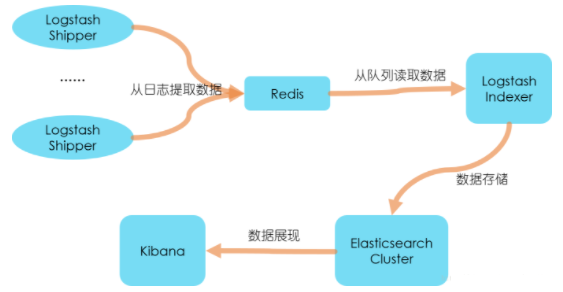
   它可以多维度的展示es中的数据。这也解决了用mysql存储带来了难以可视化的问题。他提供了丰富的UI组件，简化了使用难度，数据在es中的展示是比较让人蛋疼的，后面再讲es的时候让大家看看。

目前ELK主要有两种框架：

1.普通框架



2.个性化扩展框架（针对日志数据需要二次处理以及多方使用的场景）



三、**如何快速搭建一套稳定的ELK环境**

1.官方下载 :https://www.elastic.co/cn/products

2.版本：elasticsearch 5.6.4，kibana5.2.0 ，logstash5.6.3三个版本（注意 JDK需要1.8）

3.安装logstash

（1）解压，在config目录建：logstash.conf

（2）做好input ，filter，output三大块， 其中input是吸取logs文件下的所有log后缀的日志文件，filter是一个过滤函数，配置则可进行个性化过滤，output配置了导入到

hosts为127.0.0.1:9200的elasticsearch中，每天一个索引

|  |
| --- |
| input {  file {  type => "log"  path => "/apps/svr/servers/logs/\*.log"  start\_position => "beginning"  }  }  output {  stdout {  codec => rubydebug { }  }  elasticsearch {  hosts => "127.0.0.1"  index => "log-%{+YYYY.MM.dd}"  }  } |

说明：start\_position是监听的位置，默认是end，即一个文件如果没有记录它的读取信息，则从文件的末尾开始读取，也就是说，仅仅读取新添加的内容。对于一些更新的日志类型的监听，通常直接使用end就可以了；相反，beginning就会从一个文件的头开始读取。但是如果记录过文件的读取信息，则不会从最开始读取。重启读取信息不会丢失。

（3）bin目录下启动logstash了，配置文件设置为conf/logstash.conf

启动命令：./logstash -f ../config/logstash.conf

（4）配置多个文件：./logstash -f ../config 指定启动目录，然后启动目录下配置多个\*.Conf文件。里面指定不同的logpath。

4.安装elasticSearch

ELK中的核心，启动的时候一定要注意，因为es不可以进行root账户启动，所以你还需要重新添加一个账户，我这里是apps。在安装过程中大家会遇到以下几个问题：

（1）max file descriptors [4096] for elasticsearch process is too low, increase to at least [65536]

解决方法：vi /etc/security/limits.conf

user hard nofile 65536

user soft nofile 65536

（2）max number of threads [1024] for user [apps] is too low, increase to at least [2048]

解决办法：vi /etc/security/limits.d/90-nproc.conf

修改如下内容：

\* soft nproc 1024

#修改为

\* soft nproc 2048

（3）max virtual memory areas vm.max\_map\_count [65530] likely too low, increase to at least [262144]

解决办法：vi /etc/sysctl.conf

添加下面配置：

vm.max\_map\_count=655360

并执行命令：

sysctl -p

5.安装kibana

这个安装比较简单，解压后在kibana.yml文件中指定一下你需要读取的elasticSearch地址和可供外网访问的bind地址就可以了

elasticsearch.url: <http://localhost:9200>,如果是集群则配置master节点。  
server.host: 0.0.0.0

启动命令：./kibana

接下来我们在本机的/logs文件夹下创建一个简单的1.log文件，内容为“hello world”，然后在kibana上将logstash-\*  改成 log\* ，Create按钮就会自动出来。

echo 'hello world' > 1.log

6.安装elastic-head插件

<https://github.com/mobz/elasticsearch-head/archive/master.zip>

unzip elasticsearch-head-master.zip

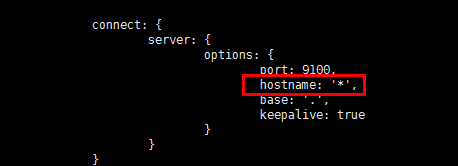
curl --silent --location https://rpm.nodesource.com/setup | bash -

yum install -y nodejs

npm install grunt --save-dev

npm install

进入elasticsearch-head-master 文件夹下，执行命令vi Gruntfile.js文件：增加hostname属性，设置为\*。



http.cors.enabled: true

http.cors.allow-origin: "\*"

./grunt server &

课外任务：部署一个head插件

同学关心较多问题：FileBeat和Logstash如何集成：

filebeat作为轻量级的logs shipper，帮助用户将无数client端上的log文件以一种轻量级的方式转发并集中日志和文件到ELK stack中，不能很好的支持过滤等个性化需求。一般可采取fileBeat收集服务器日志，发送到Logstash中进行果过滤

集成：安装好fileBeat后只需要在logstash上面修改下input，改为从fileBeat读取即可。

input {

beats {

port => filebeat的端口号，默认为5044

}

}