# ELK课件

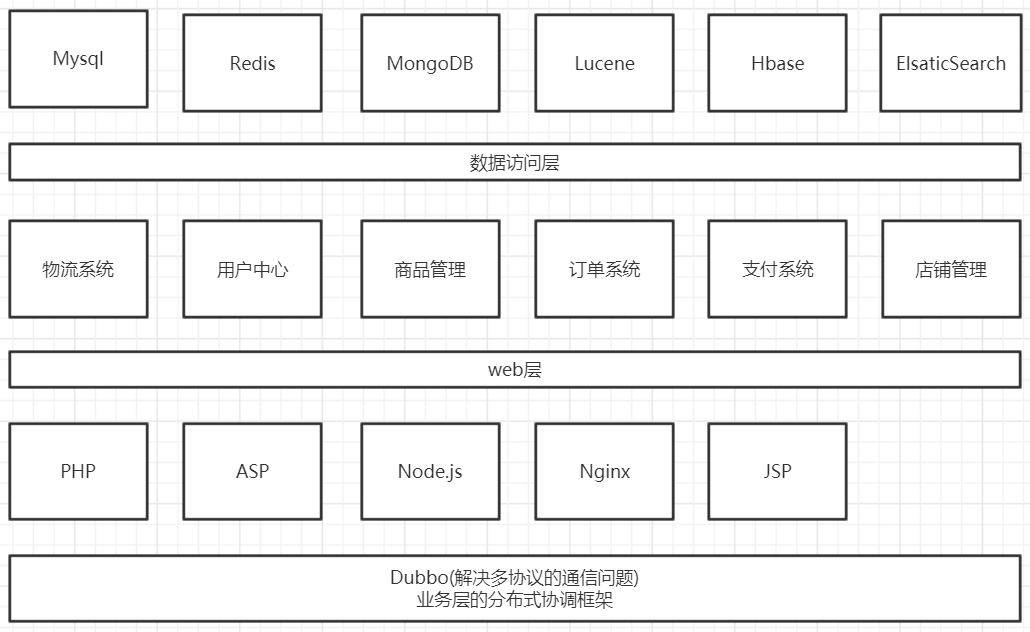
## ELK简介

### ELK是什么

* 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。
* Logstash是一个完全开源的工具，他可以对你的日志进行收集、过滤，并将其存储供以后使用（如，搜索）。
* Kibana 也是一个开源和免费的工具，它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。

### ELK应用

* 电商体系架构

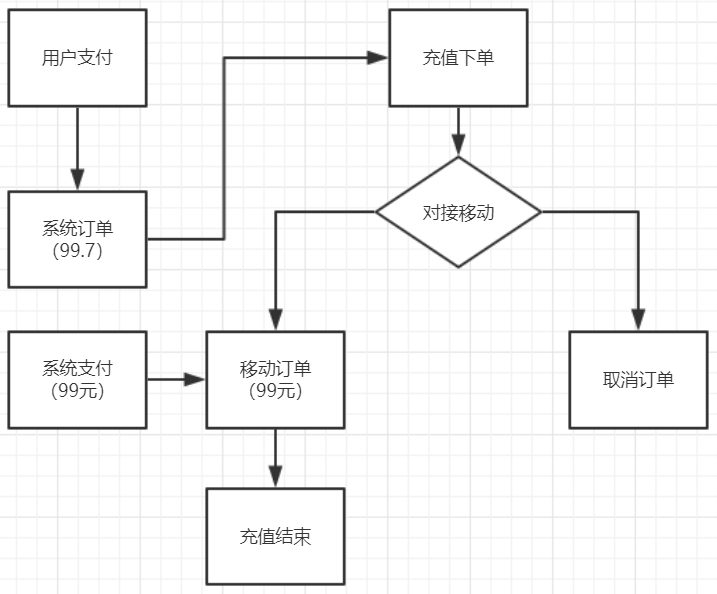


* 问题

1. API不一样，我们如何去整合？--》dubbo 定义统一的api规范
2. 各子系统之间会产生操作痕迹（用户行为轨迹）---》日志
3. 各个子系统都会生成各自的日志---日志整合--》logstash
4. AOP埋点，异步日志输出

* 具体场景1

通过第三方进行移动话费充值



日志输出：每次调用都会打印异步日志

分布式负载均衡：

很多太机器都可以充值（动态的去选择一台目前比较空闲的机器去执行这个任务）

问题：

A：兄弟，帮忙查一下今天手机号码138001380000充值日志记录（是否成功）

B：稍等

5分钟后

A：怎么样了

B：稍等，还剩下3台机器没查完

结论：如果能把所有的日志整理在一起，就不会出现一台一台去查的问题

解决方案：

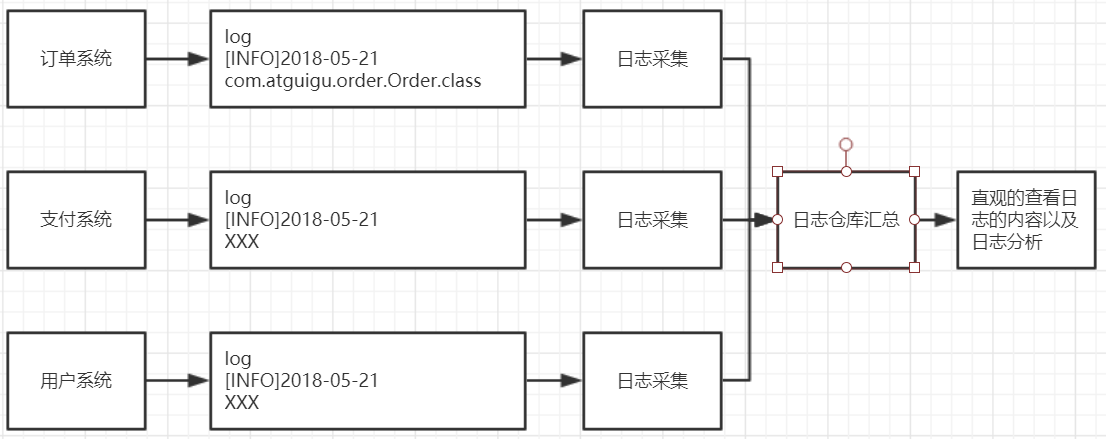
1. 可不可以把日志放在数据库中。

数据量太大，且日志没有规范日志格式，数据库方案不太建议，且压力过大

1. 采用大数据日志处理方案

成本太高，且分布式环境每个系统的日志规则不一样。

* 具体业务实践



日志收集：Logstash

日志存储：ElasticSearch

日志展示：Kibana

针对对台服务器日志不统一的问题，提供多种检索规则，方便可视化展示

* 案例总结

分布式带来的问题：多节点、负载均衡、日志分散、运维成本高（需要人为跟踪）

### 集中式日志管理系统

当前主流的一些集中日志管理系统

1. 简单的：Rsyslog
2. 商业化：Splunk
3. 开源的：Scribe（FaceBook），Chukwa（Apache）
4. ELK最广泛的（Elastic Stack）(java语言编写)

[www.elastic.co](http://www.elastic.com)/cn

### ELK

|  |  |  |
| --- | --- | --- |
| ElasticSearch | Java | 实时的分布式搜索和分析引擎，他可以用于全文检索，结构化搜索以及分析，lucene。Solr |
| Logstash | JRuby | 具有实时渠道能力的数据收集引擎，包含输入、过滤、输出模块，一般在过滤模块中做日志格式化的解析工作 |
| Kibana | JavaScript | 为ElasticSerach提供分析平台和可视化的Web平台。他可以ElasticSerach的索引中查找，呼唤数据，并生成各种维度的表图 |

### 日志

日志：记录程序的运行轨迹---

级别：ERROR、INFO、DEBUG、WARN

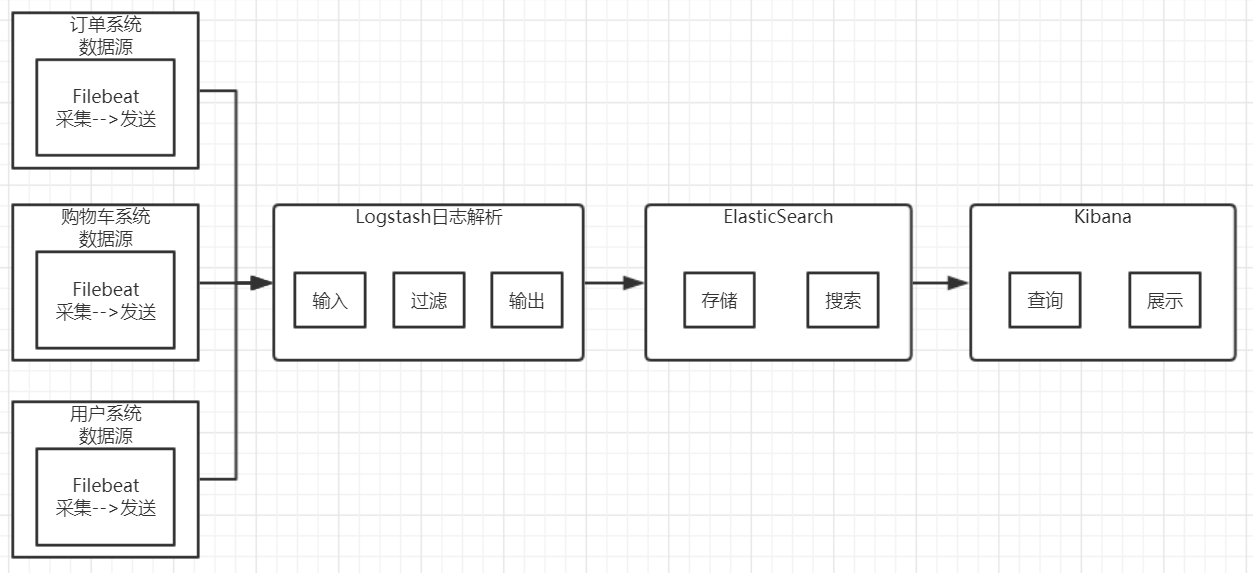
目的：方便定位和查找信息，记录除去业务外的附加的信息，链路

Filebeat简介

当您要面对成百上千、甚至成千上万的服务器、虚拟机和容器生成的日志时，请告别 SSH 吧。Filebeat 将为您提供一种轻量型方法，用于转发和汇总日志与文件，让简单的事情不再繁杂。

当将数据发送到 Logstash 或 Elasticsearch 时，Filebeat 使用背压敏感协议，以考虑更多的数据量。如果 Logstash 正在忙于处理数据，则可以让 Filebeat 知道减慢读取速度。一旦拥堵得到解决，Filebeat 就会恢复到原来的步伐并继续运行。

无论在任何环境中，随时都潜伏着应用程序中断的风险。Filebeat 能够读取并转发日志行，如果出现中断，还会在一切恢复正常后，从中断前停止的位置继续开始。



## 2、准备工作

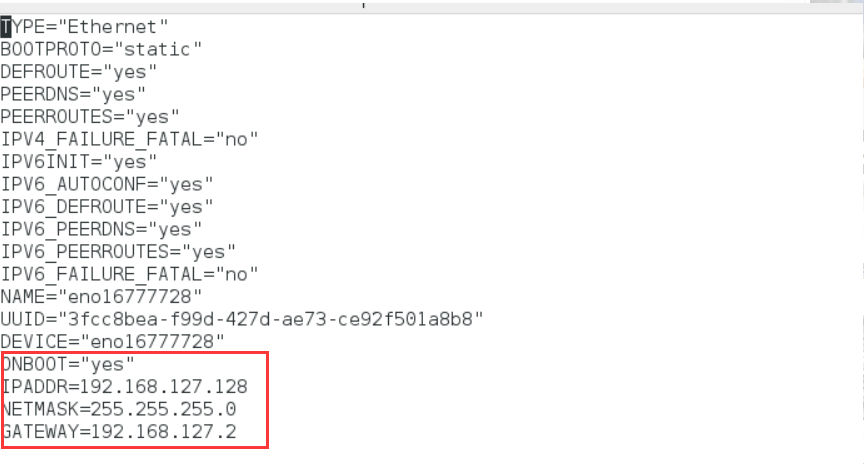
### 2.1、安装Centos7

建议内存2G以上

### 2.2、基本配置

* 设置IP地址

vi /etc/sysconfig/network-scripts/ifcfg-eno33



service network restart

* 添加用户并授权

[root@localhost ~]# adduser elk1

[root@localhost ~]# passwd elk1

[root@localhost ~]# whereis sudoers

[root@localhost ~]# ls -l /etc/sudoers

[root@localhost ~]# chmod -v u+w /etc/sudoers

[root@localhost ~]# vi /etc/sudoers

## Allow root to run any commands anywher

root ALL=(ALL) ALL

linuxidc ALL=(ALL) ALL #这个是新增的用户

[root@localhost ~]# chmod -v u-w /etc/sudoers

[root@localhost ~]# su elk1

## ElasticSerach

### 3.1、Java环境安装

* 解压安装包

[root@localhost jdk1.8]# tar -zxvf jdk-8u171-linux-x64.tar.gz

* 设置Java环境变量

[root@localhost jdk1.8.0\_171]# vi /etc/profile

在文件最后添加

export JAVA\_HOME=/home/elk1/jdk1.8/jdk1.8.0\_171

export JRE\_HOME=$JAVA\_HOME/jre

export CLASSPATH=.:$JAVA\_HOME/LIB:$JRE\_HOME/LIB:$CLASSPATH

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

[root@localhost jdk1.8.0\_171]# source /etc/profile

[root@localhost jdk1.8.0\_171]# java -version

java version "1.8.0\_171"

Java(TM) SE Runtime Environment (build 1.8.0\_171-b11)

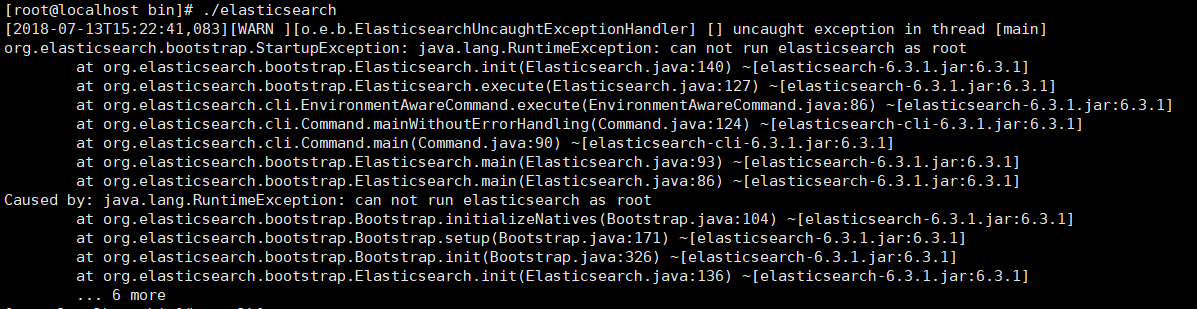
Java HotSpot(TM) 64-Bit Server VM (build 25.171-b11, mixed mode)

### 3.2、ElasticSerach单机安装

[root@localhost elasticserach]# tar -zxvf elasticsearch-6.3.1.tar.gz

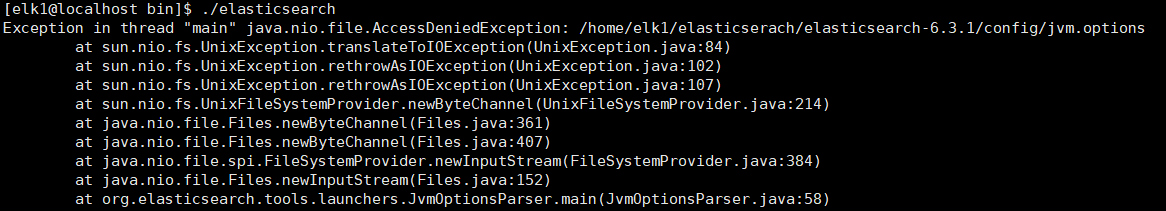
[root@localhost elasticserach]# cd elasticsearch-6.3.1/bin

[root@localhost bin]# ./elasticsearch



[root@localhost bin]# su elk1

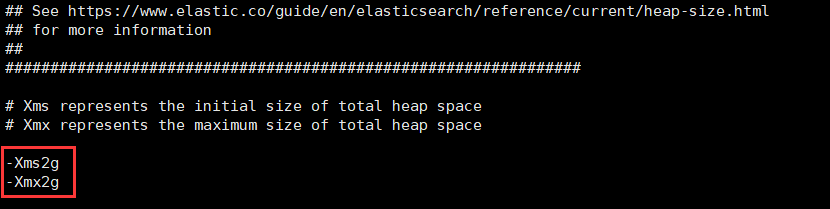
[elk1@localhost bin]$ ./elasticsearch



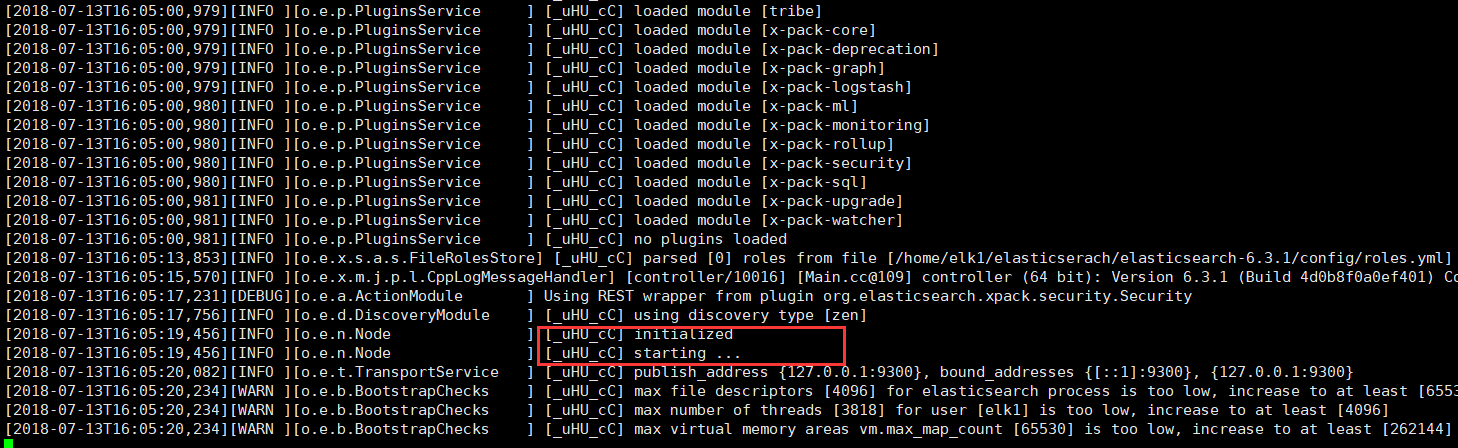
[root@localhost bin]# chown -R elk1:elk1 /home/elk1/elasticsearch

[elk1@localhost bin]$ ./elasticsearch

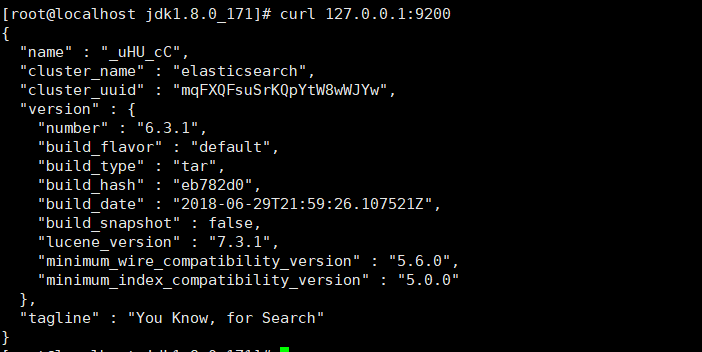
[elk1@localhost config]$ vi jvm.options



[elk1@localhost bin]$ ./elasticsearch



[root@localhost jdk1.8.0\_171]# curl 127.0.0.1:9200

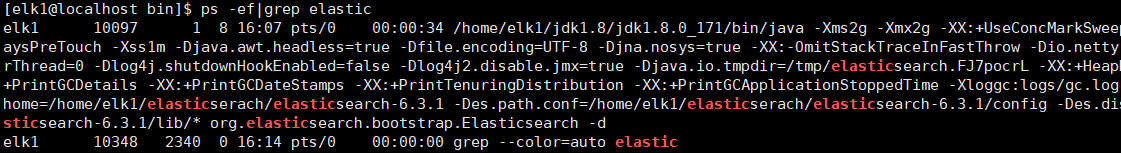


#后台启动

[elk1@localhost bin]$ ./elasticsearch -d

#关闭程序

[elk1@localhost bin]$ ps -ef|grep elastic

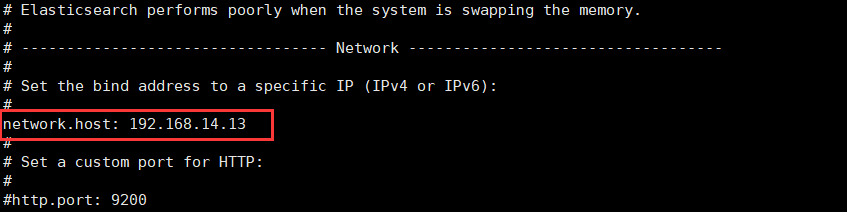


[elk1@localhost bin]$ kill 10097

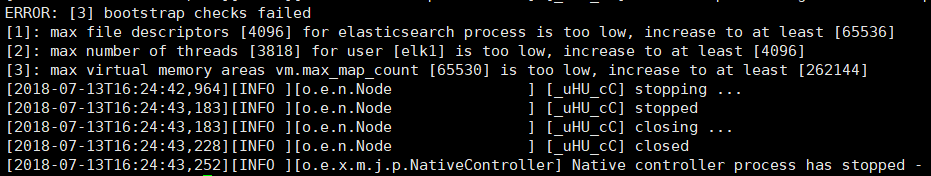
#设置浏览器访问

[root@localhost bin]systemctl stop firewalld

[root@localhost bin]vi config/elasticsearch.yml



安装问题：



1. [2]解决方案

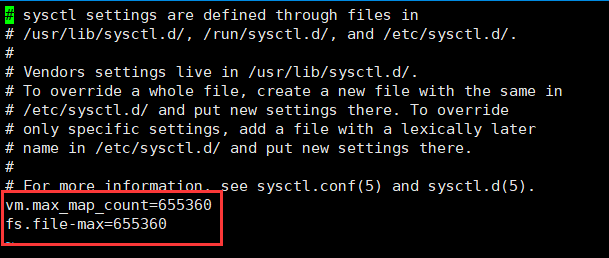
[root@localhost bin]# vi /etc/security/limits.conf



1. 解决方案

[root@localhost bin]# vi /etc/sysctl.conf

[root@localhost bin]# sysctl -p



### 3.3、ElasticSerach集群安装

* 修改配置文件elasticserach.yml

vim /elasticsearch.yml

cluster.name: aubin-cluster#必须相同

# 集群名称（不能重复）

node.name: els1（必须不同）

# 节点名称，仅仅是描述名称，用于在日志中区分（自定义）

#指定了该节点可能成为 master 节点，还可以是数据节点  
 node.master: true  
 node.data: true

path.data: /var/lib/elasticsearch

# 数据的默认存放路径（自定义）

path.logs: /var/log/elasticsearch

# 日志的默认存放路径

network.host: 192.168.0.1

# 当前节点的IP地址

http.port: 9200

# 对外提供服务的端口

transport.tcp.port: 9300

#9300为集群服务的端口

discovery.zen.ping.unicast.hosts: ["172.18.68.11", "172.18.68.12","172.18.68.13"]

# 集群个节点IP地址，也可以使用域名，需要各节点能够解析

discovery.zen.minimum\_master\_nodes: 2

# 为了避免脑裂，集群节点数最少为 半数+1

注意：清空data和logs数据

192.168.14.12:9200/\_cat/nodes?v

### 3.4、安装head插件

* 下载head插件

wget [https://github.com/mobz/elasticsearch-head/archive/elasticsearch-head-master.zip](https://github.com/mobz/elasticsearch-head/archive/master.zip)

也可以用git下载，前提yum install git

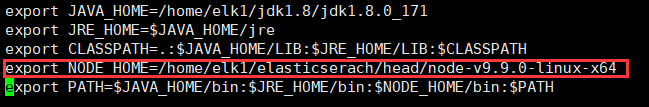
unzip elasticsearch-head-master.zip

* 安装node.js

wget <https://npm.taobao.org/mirrors/node/latest-v4.x/node-v4.4.7-linux-x64.tar.gz>

tar -zxvf node-v9.9.0-linux-x64.tar.gz

* 添加node.js到环境变量



source /etc/profile

* 测试

node -v

npm -v

* 安装grunt（grunt是一个很方便的构建工具，可以进行打包压缩、测试、执行等等的工作）

进入到elasticsearch-head-master

npm install -g grunt-cli

npm install

(npm install -g cnpm --registry=https://registry.npm.taobao.org)

* 修改Elasticsearch配置文件

编辑elasticsearch-6.3.1/config/elasticsearch.yml,加入以下内容：

http.cors.enabled: true

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

* 修改Gruntfile.js（注意’，’）

打开elasticsearch-head-master/Gruntfile.js，找到下面connect属性，新增hostname:’\*’:

connect: {

server: {

options: {

hostname: '\*',

port: 9100,

base: '.',

keepalive: true

}

}

}

* 启动elasticsearch-head

进入elasticsearch-head目录，执行命令：grunt server

* 后台启动elasticsearch-head

nohup grunt server &exit

* 关闭head插件

ps -aux|grep head

kill 进程号

### 3.5、ElasticSerach API

* elasticsearch rest api遵循的格式为：

curl -X<REST Verb> <Node>:<Port>/<Index>/<Type>/<ID>

* 检查es版本信息

curl IP:9200

* 查看集群是否健康

<http://IP:9200/_cat/health?v>

* 查看节点列表

<http://IP:9200/_cat/nodes?v>

* 列出所有索引及存储大小

<http://IP:9200/_cat/indices?v>

* 创建索引

curl -XPUT 'IP:9200/XX?pretty'

* 添加一个类型

curl -XPUT 'IP:9200/XX/external/2?pretty' -d '

{

   "gwyy": "John"

}'

* 更新一个类型

curl -XPOST 'IP:9200/XX/external/1/\_update?pretty' -d '

{

   "doc": {"name": "Jaf"}

}'

* 删除指定索引

curl -XDELETE 'IP:9200/\_index?pretty'

### 3.6、配置详情

* ElasticSearch.yml

ES的相关配置

# 集群的名字，以此作为是否同一集群的判断条件

cluster.name: elasticsearch

# 节点名字，以此作为集群中不同节点的区分条件

node.name: node-1

#设置当前节点既可以为主节点也可以为数据节点

node.master: true

node.data: true

# 索引分片个数，默认为5片

#index.number\_of\_shards: 5

# 索引副本个数，默认为1个副本

#index.number\_of\_replicas: 1

# 数据存储目录（多个路径用逗号分隔）

discovery.zen.ping.unicast.hosts: ["192.168.14.14","192.168.14.15"]

discovery.zen.minimum\_master\_nodes: 2

#数据目录

path.data: /home/elk1/elasticserach/data

# 日志目录

path.logs: /home/elk1/elasticserach/logs

# 修改一下ES的监听地址，这样别的机器才可以访问

network.host: 192.168.14.13

# 设置节点间交互的tcp端口（集群）,默认是9300

transport.tcp.port: 9300

# 监听端口（默认的就好）

http.port: 9200

# 增加新的参数，这样head插件才可以访问es

http.cors.enabled: true

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

* Jvm.options

JVM的相关配置

* Log4j2.properties

日志相关配置

### 3.7、Elasticserach模式

* 分为Development和Production两种模式
* 区分方式

以transport的地址是否绑定在localhost为标准（实际地址）

即：elasticserach.yml文件中的network.host配置

* 模式区别

1. Development模式下启动时会以warning的方式提示配置检查异常
2. Production模式下在启动时会以error的方式提示配置检查异常并推出

### 3.8、elasticserach操作

* 基本概念
* Document:文档对象
* Index:索引（库）
* Type:索引中的数据类型（表）
* Field:字段，文档的属性（字段）
* Query DSL:查询语法（sql）
* CRUD操作
* 创建文档

请求：

POST /atguigu/student/1

{

"name":"zhangsan",

"clazz":"0115bigdata",

"description":"we are family"

}

返回：

{

"\_index": "atguigu",

"\_type": "student",

"\_id": "1",

"\_version": 1,

"result": "created",

"\_shards": {

"total": 2,

"successful": 2,

"failed": 0

},

"\_seq\_no": 0,

"\_primary\_term": 1

}

* 获取文档

请求：

GET atguigu/student/1

返回：

{

"\_index": "atguigu",

"\_type": "student",

"\_id": "1",

"\_version": 1,

"found": true,

"\_source": {

"name": "zhangsan",

"clazz": "0115bigdata",

"description": "we are family"

}

}

* 更新文档

请求：

POST /atguigu/student/1/\_update

{

"doc":{

"description":"hello world"

}

}

返回：

{

"\_index": "atguigu",

"\_type": "student",

"\_id": "1",

"\_version": 2,

"result": "updated",

"\_shards": {

"total": 2,

"successful": 2,

"failed": 0

},

"\_seq\_no": 1,

"\_primary\_term": 1

}

* 删除文档

请求：

DELETE atguigu/student/1

查询结果：

{

"\_index": "atguigu",

"\_type": "student",

"\_id": "1",

"found": false

}

* Elasticserach Query
* Query String

GET /atguigu/student/\_sea'rch?q=关键字

返回：

{

"took": 8,

"timed\_out": false,

"\_shards": {

"total": 5,

"successful": 5,

"skipped": 0,

"failed": 0

},

"hits": {

"total": 1,

"max\_score": 0.2876821,

"hits": [

{

"\_index": "atguigu",

"\_type": "student",

"\_id": "1",

"\_score": 0.2876821,

"\_source": {

"name": "zhangsan",

"clazz": "0115bigdata",

"description": "we are family"

}

}

]

}

}

* Query DSL

GET atguigu/student/\_search

{

"query":{

"term":{

"name":{

"value":"zhangsan"

}

}

}

}

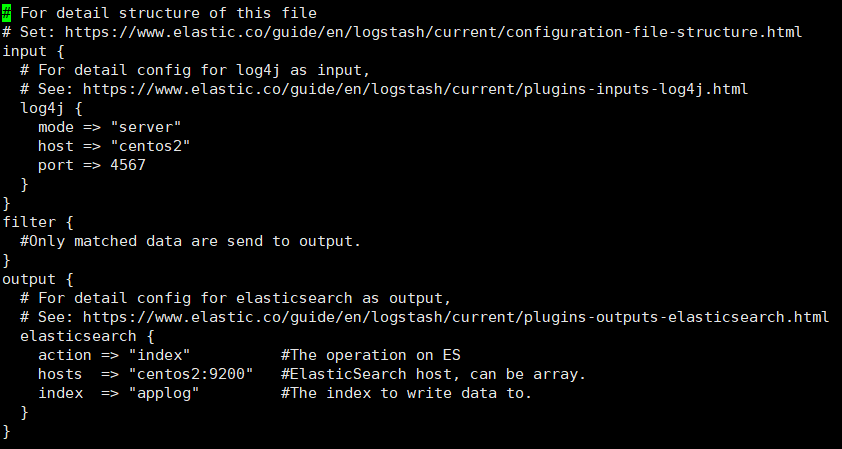
## Logstash

### 4.1、安装logstash

[root@localhost logstash]# tar -zxvf logstash-6.3.1.tar.gz

[root@localhost logstash-6.3.1]# cd config

[root@localhost config]# vi log4j\_to\_es.conf



input {

file {

path=>[""]

type=>""

start\_position=>"beginning"

}

}

output {

stdout {

codec=>rubydebug

}

}

[root@localhost logstash-6.3.1]# ./bin/logstash -f config/log4j\_to\_es.conf

### 4.2、输入、输出、过滤

* 输入

input{file{path=>”/tomcat/logs/abc.log”}}

* 输出

output{stdout{codec=>rubydebug}}

* 过滤插件
* Grok

1. 基于正则表达式提供了丰富可重用的模式（pattern）
2. 基于此可以将非结构化数据作结构化处理

* Date

将字符串类型的时间字段转换为时间戳类型，方便后续数据处理

* Mutate

进行增加、修改、删除、替换等字段相关处理

### 4.3、logstash格式化nginx日志内容

* 创建nginx\_logstash.conf文件

input {

stdin { }

}

filter {

grok {

match => {

"message" => '%{IPORHOST:remote\_ip} - %{DATA:user\_name} \[%{HTTPDATE:time}\] "%{WORD:request\_action} %{DATA:request} HTTP/%{NUMBER:http\_version}" %{NUMBER:response} %{NUMBER:bytes} "%{DATA:referrer}" "%{DATA:agent}"'

}

}

date {

match => [ "time", "dd/MMM/YYYY:HH:mm:ss Z" ]

locale => en

}

geoip {

source => "remote\_ip"

target => "geoip"

}

useragent {

source => "agent"

target => "user\_agent"

}

}

output {

stdout {

codec => rubydebug

}

}

* Logstash启动解析nginx文件

head -n 2 /home/elk1/nginx\_logs|./logstash -f ../config/nginx\_logstash.conf

* 结果

{

"user\_name" => "-",

"referrer" => "-",

"@timestamp" => 2015-05-17T08:05:32.000Z,

"request" => "/downloads/product\_1",

"time" => "17/May/2015:08:05:32 +0000",

"geoip" => {

"country\_code3" => "NL",

"longitude" => 4.8995,

"continent\_code" => "EU",

"latitude" => 52.3824,

"timezone" => "Europe/Amsterdam",

"country\_code2" => "NL",

"ip" => "93.180.71.3",

"country\_name" => "Netherlands",

"location" => {

"lat" => 52.3824,

"lon" => 4.8995

}

},

"@version" => "1",

"http\_version" => "1.1",

"remote\_ip" => "93.180.71.3",

"message" => "93.180.71.3 - - [17/May/2015:08:05:32 +0000] \"GET /downloads/product\_1 HTTP/1.1\" 304 0 \"-\" \"Debian APT-HTTP/1.3 (0.8.16~exp12ubuntu10.21)\"",

"bytes" => "0",

"user\_agent" => {

"minor" => "3",

"os" => "Debian",

"name" => "Debian APT-HTTP",

"os\_name" => "Debian",

"build" => "",

"major" => "1",

"device" => "Other"

},

"agent" => "Debian APT-HTTP/1.3 (0.8.16~exp12ubuntu10.21)",

"host" => "localhost.localdomain",

"response" => "304",

"request\_action" => "GET"

}

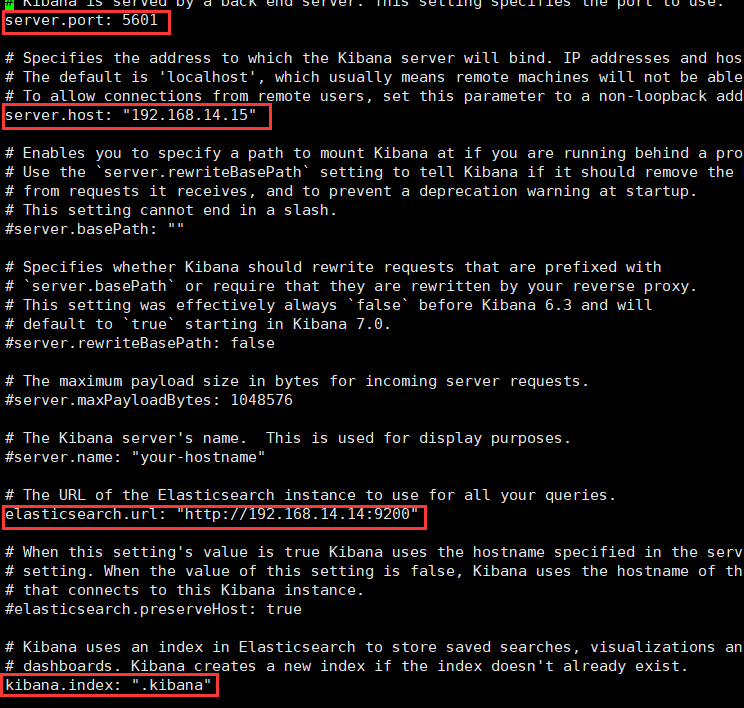
## Kibana

### 5.1、Kibana安装

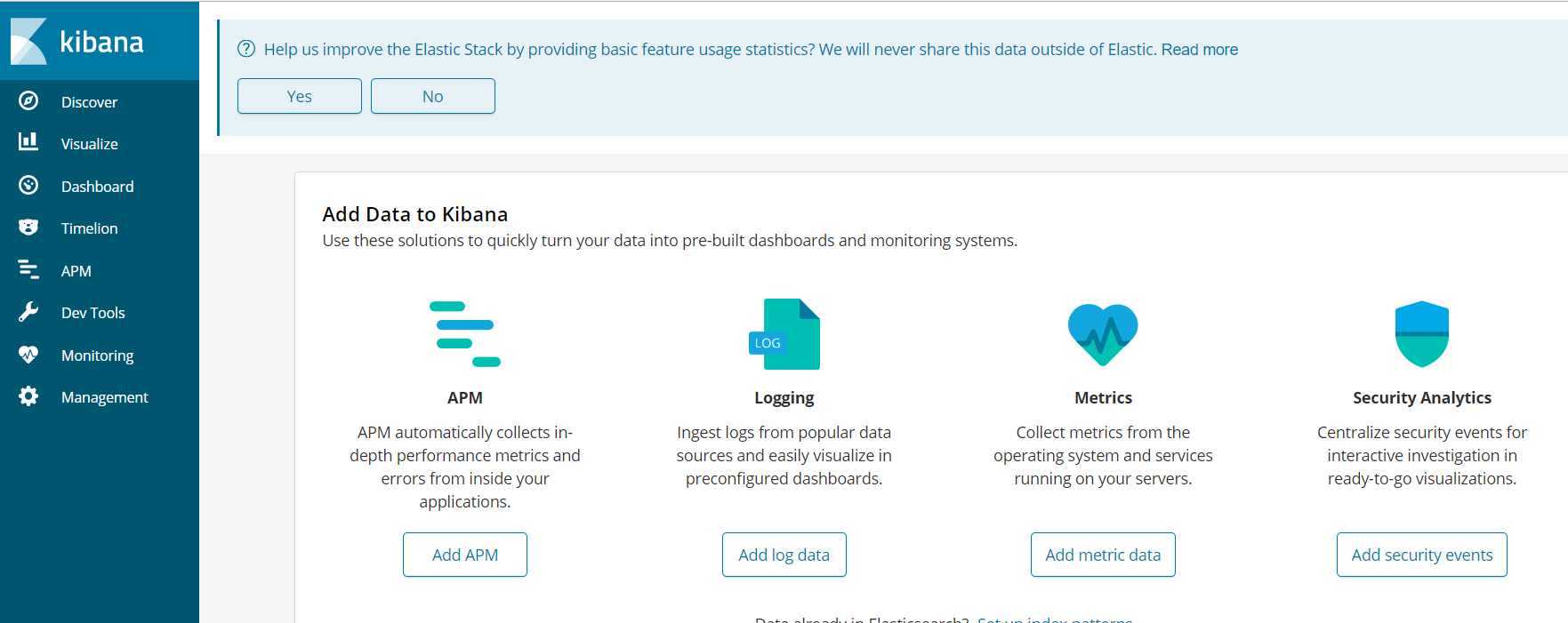
[root@localhost kibana]# tar -zxvf kibana-6.3.1-linux-x86\_64.tar.gz

[root@localhost kibana]# cd kibana-6.3.1-linux-x86\_64/config

[root@localhost config]# vi kibana.yml

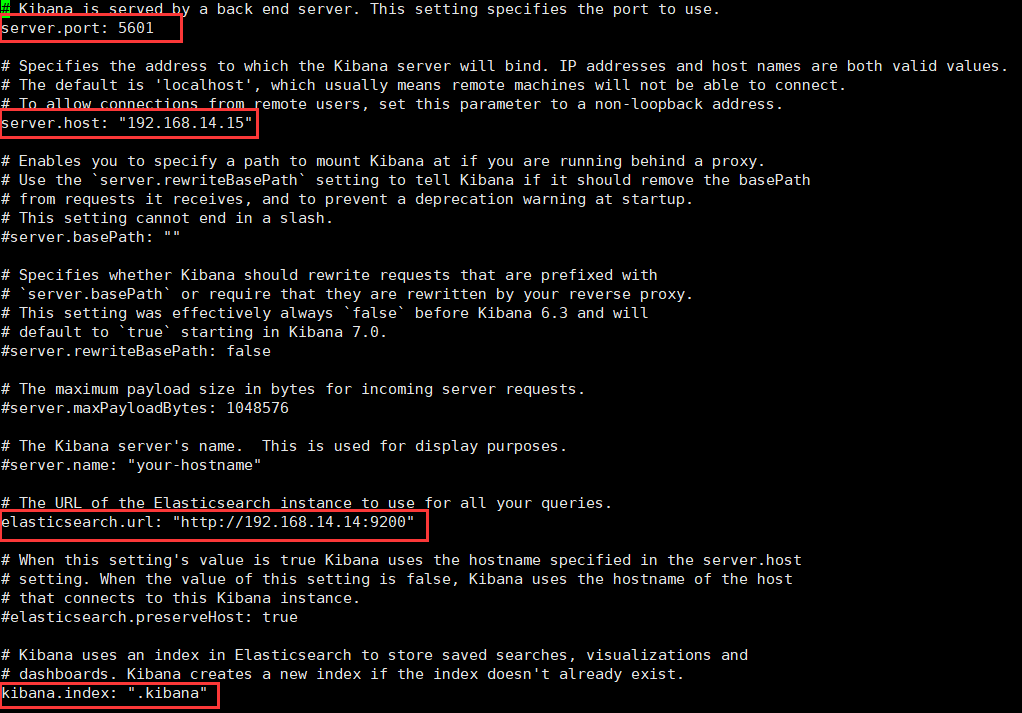


[root@localhost bin]# ./kibana



### 5.2、kibana配置

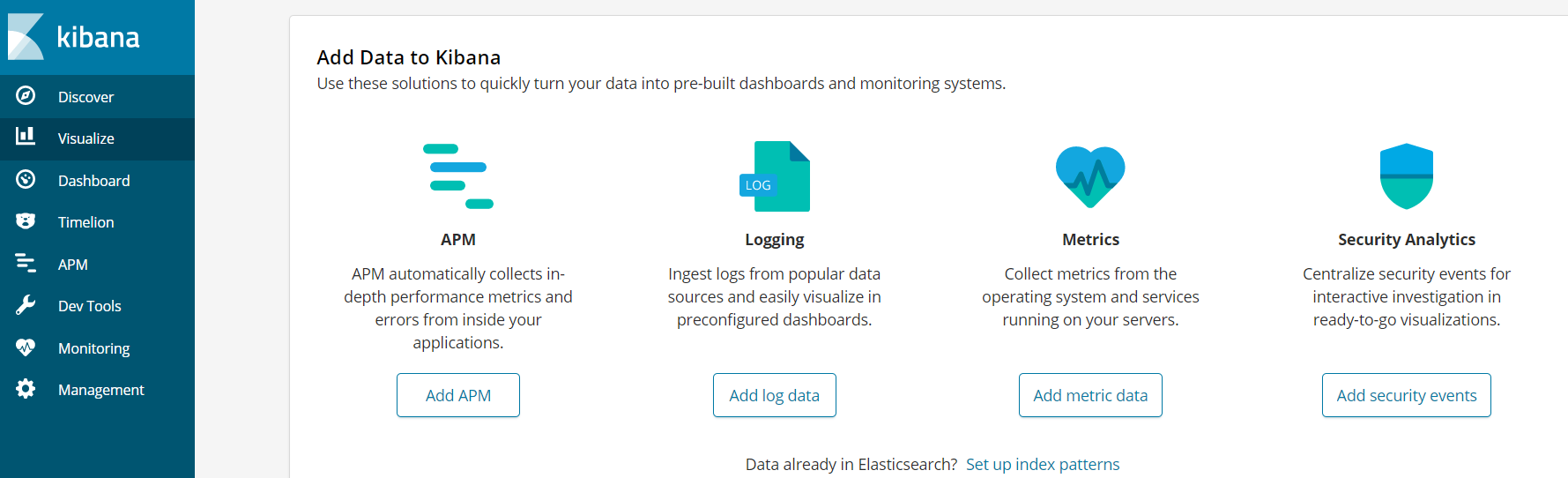
* 配置文件在config文件夹下
* Kibana.yml常用配置说明



Server.host/server.port:访问的端口号和地址(地址设置后才能被外网访问)

Elasticsearch.url:访问elasticserach的地址

### 5.3、kibana功能简介



Discover:数据搜索查看

Visualize:图标制作

Dashboard:仪表盘制作

Timeline:时序数据的高级可视化分析

DevTools:开发者工具

Management:kibana相关配置

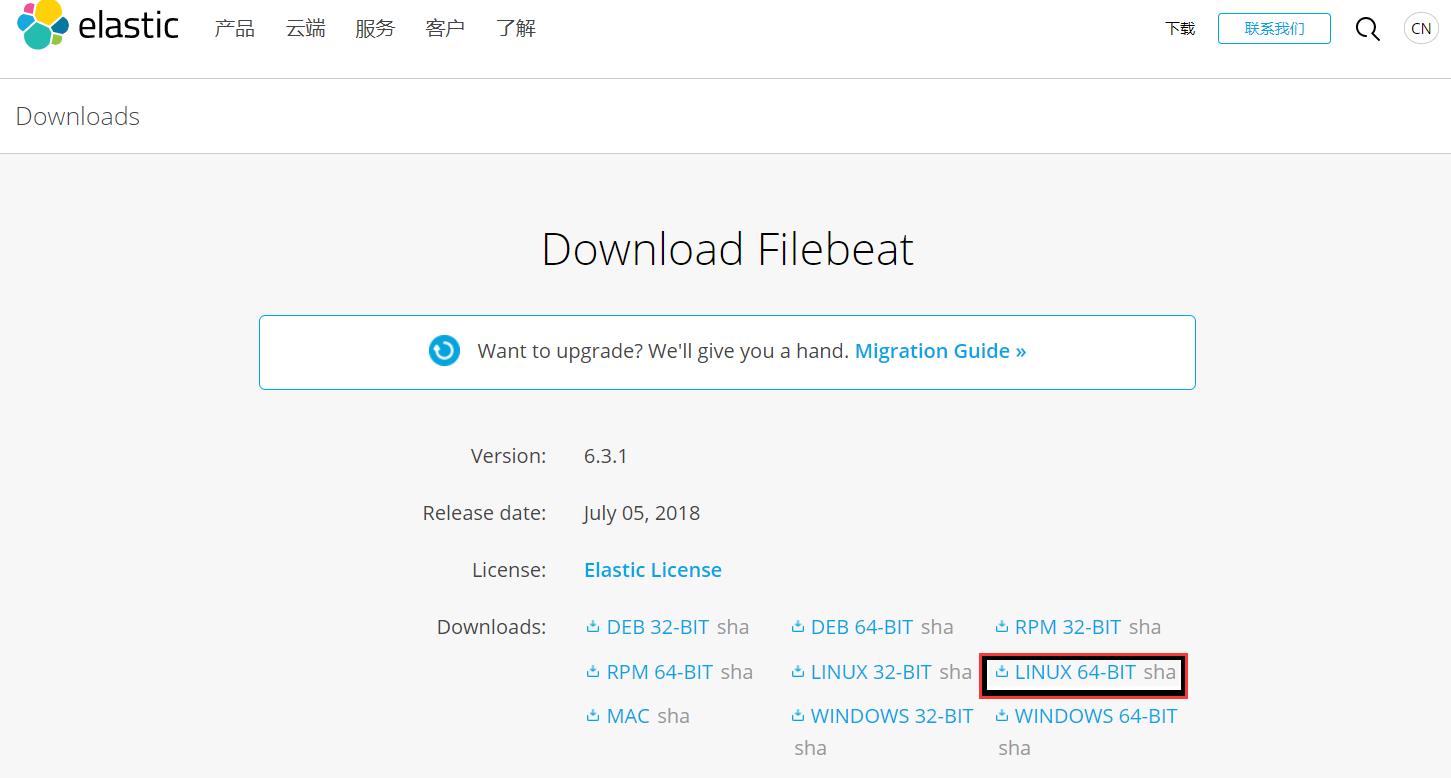
## 6、Filebeat和packetbeat

### 2.1、Filebeat

* 下载Filebeat

<https://www.elastic.co/cn/downloads/beats/filebeat>

查看系统位数：getconf LONG\_BIT



### 2.2、Packetbeat

* Packetbeat简介

（1）实时抓取网络包

（2）自动解析应用层协议（抓包）

DNS、Http、Redis、Mysql等

* Packetbeat抓取elasticserach请求数据

1. 进入packetbeat目录，创建es.yml文件
2. 编辑es.yml文件

packetbeat.interfaces.device: ens33#网卡

packetbeat.protocols.http:

ports: [9200]#es端口

send\_request: true#抓取请求信息

include\_body\_for: ["application/json", "x-www-form-urlencoded"]#包含内容

output.console:

pretty: true#控制台输出

1. 启动packetbeat

sudo ./packetbeat -e -c es.yml -strict.perms=false

## 7、Nginx

* 安装nginx

#安装依赖环境

yum install gcc-c++

yum install pcre-devel

yum install zlib zlib-devel

yum install openssl openssl-deve

#//一键安装上面四个依赖

#yum -y install gcc zlib zlib-devel pcre-devel openssl openssl-devel

#解压

tar -xvf nginx-1.13.7.tar.gz

#进入nginx目录

cd /usr/local/nginx #执行命令

./configure

#执行make命令make//执行make install命令

make

make install

//启动命令

nginx/sbin/nginx

//停止命令

nginx/sbin/nginx -s stop或者 : nginx -s quit  
 //重启命令  
 nginx -s reload

## 8、数据可视化演示实战

### 8.1、实战说明

* 需求：

收集Elasticserach集群的查询语句

分析查询语句的常用语句、响应时长等

* 方案

数据收集：Packetbeat+logstash

数据分析：Kibana+Elasticsearch

### 8.2、前期准备

* Production Cluster(生产环境)

1. Elasticsearch 192.168.14.13:9200
2. Kibana 192.168.14.15:5601

* Monitoring Cluster(监控环境)

1. Elasticsearch 192.168.14.16:8200
2. Kibana 192.168.14.16:8601

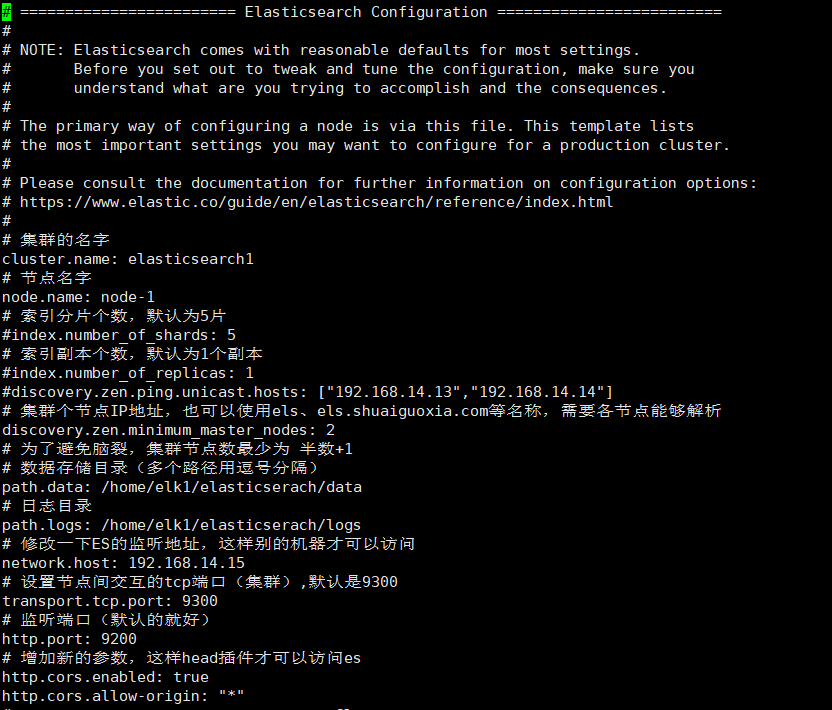
* Logstash\packetbeat

### 8.3、实战

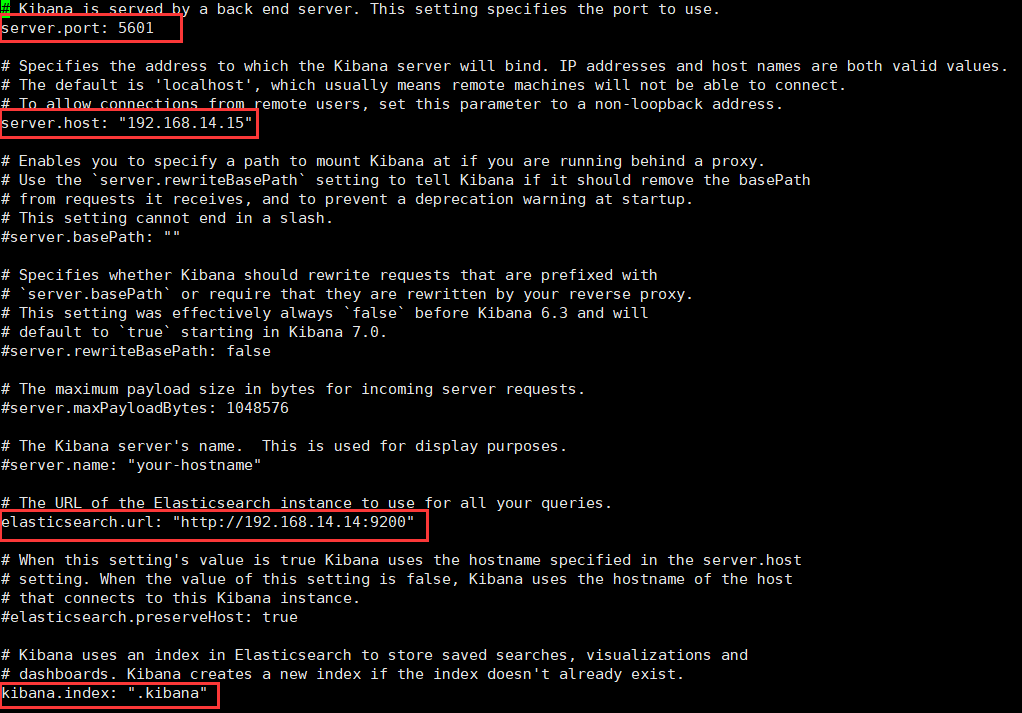
* 启动数据采集集群

启动ES：

./elasticsearch



修改kibana配置



./kibana #启动

* 启动数据分析集群

1. 启动ES

同上

1. 启动logstash

input {

beats {

port => 5044

}

}

filter {

if "search" in [request]{

grok {

match => { "request" => ".\*\n\{(?<query\_body>.\*)"}

}

grok {

match => { "path" => "\/(?<index>.\*)\/\_search"}

}

if [index] {

} else {

mutate {

add\_field => { "index" => "All" }

}

}

mutate {

update => { "query\_body" => "{%{query\_body}"}}

}

# mutate {

# remove\_field => [ "[http][response][body]" ]

# }

}

output {

#stdout{codec=>rubydebug}

if "search" in [request]{

elasticsearch {

hosts => "127.0.0.1:9200"

}

}

}

1. 启动

./bin/logstash -f config/log4j\_to\_es.conf

### 附录：防火墙配置

1、firewalld的基本使用

启动： systemctl start firewalld

关闭： systemctl stop firewalld

查看状态： systemctl status firewalld

开机禁用  ： systemctl disable firewalld

开机启用  ： systemctl enable firewalld

2.systemctl是CentOS7的服务管理工具中主要的工具，它融合之前service和chkconfig的功能于一体。

启动一个服务：systemctl start firewalld.service  
 关闭一个服务：systemctl stop firewalld.service  
 重启一个服务：systemctl restart firewalld.service  
 显示一个服务的状态：systemctl status firewalld.service  
 在开机时启用一个服务：systemctl enable firewalld.service  
 在开机时禁用一个服务：systemctl disable firewalld.service  
 查看服务是否开机启动：systemctl is-enabled firewalld.service  
 查看已启动的服务列表：systemctl list-unit-files|grep enabled  
 查看启动失败的服务列表：systemctl --failed

3.配置firewalld-cmd

查看版本： firewall-cmd --version

查看帮助： firewall-cmd --help

显示状态： firewall-cmd --state

查看所有打开的端口： firewall-cmd --zone=public --list-ports

更新防火墙规则： firewall-cmd --reload

查看区域信息:  firewall-cmd --get-active-zones

查看指定接口所属区域： firewall-cmd --get-zone-of-interface=eth0

拒绝所有包：firewall-cmd --panic-on

取消拒绝状态： firewall-cmd --panic-off

查看是否拒绝： firewall-cmd --query-panic

4.那怎么开启一个端口呢

添加

firewall-cmd --zone=public --add-port=80/tcp --permanent    （--permanent永久生效，没有此参数重启后失效）

重新载入

firewall-cmd --reload

查看

firewall-cmd --zone= public --query-port=80/tcp

删除

firewall-cmd --zone= public --remove-port=80/tcp --permanent