ELK

# 简介

**Elasticsearch + Logstash + Kibana（ELK）是一套开源的日志管理方案，分析网站的访问情况时我们一般会借助Google/百度等方式嵌入JS做数据统计，但是当网站访问异常或者被攻击时我们需要在后台分析如Nginx的具体日志，而Nginx日志分割/GoAccess/Awstats都是相对简单的单节点解决方案，针对分布式集群或者数据量级较大时会显得心有余而力不足，而ELK的出现可以使我们从容面对新的挑战**。

## 1.1 Elasticsearch

**Elasticsearch 是一个建立在全文搜索引擎 Apache Lucene™ 基础上的分布式的，高可用的，基于json格式的数据构建索引，准实时查询的搜索引擎。Lucene 是当今最先进最高效的全功能开源搜索引擎框架,但是Lucene使用非常复杂。**

**Elasticsearch使用 Lucene 作为内部引擎，但是在你使用它做全文搜索时，只需要使用统一开发好的API即可，而并不需要了解其背后复杂的 Lucene 的运行原理。**

## 1.2 Logstash

**logstash是一个应用程序日志、事件的传输、处理、管理和搜索的平台。你可以用它来统一对应用程序日志进行收集管理，提供 Web 接口用于查询和统计。其实logstash是可以被别的替换，比如常见的fluented.**

## 1.3 Kibana

**Kibana是一个为 Logstash 和 ElasticSearch 提供的日志分析的 Web 接口。可使用它对日志进行高效的搜索、可视化、分析等各种操作**。

# ELK安装

## 2.1 环境配置

**硬件：**

**Redhat-7.x系列\_x86\_64**

**Mem建议至少4G**

**软件：**

**elasticsearch-5.2.0**

**kibana-5.2.0-linux-x86\_64**

**logstash-5.2.0**

**redis-3.0.5（稳定版本）**

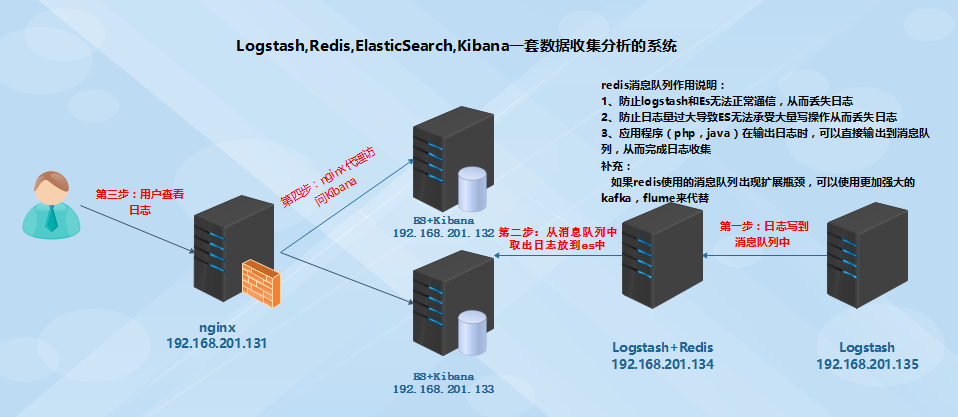
**java-1.8**

**nginx-1.10**

**安装方式采用源码**

**架构图：**

**线上环境可以这么做（根据实际环境配置）**



**简单描述下：**

**日志收集系统：（数据来源）生产者产生的日志记录行为，通过被logstash收集、转发，然后传送给redis做序列，最后通过ELK处理分析展示。**

**elasticSearch会对数据做存储，索引（基于Lunce），再kibana中建立对elasticSearch的链接，实时的抓取索索引后的数据，这样数据就可以实时的进行展示，通过一些数据组装，查询条件，得到我们想要的结果。**

**本次用两台实验**

|  |  |
| --- | --- |
| **节点** | **作用** |
| **192.168.201.137（server）** | **安装elasticsearch，kibana，logstash，redis，java** |
| **192.168.201.136（client）** | **安装nginx，java，logstash** |

**角色扮演：**

**192.168.201.137--------》用于存储数据**

**192.168.201.136--------》agent 把数据传输给192.168.201.137**

**logstash收集数据**

**redis提供队列**

**elasticsearch存储数据**

**注意：**

**es（elasticsearch）版本2.x以上需要JDK 1.8以上**

**运行es不能使用root用来来运行**

**es目录必须指定一个普通用户和组（授权）**

**es对内存和CPU的消耗比较高**

**es使用的端口看开放iptables:9200，9300等**

**es配置其他插件实现资源等可视化监控**

**es的版本和插件之间版本要匹配**

**es集群配置，第一节点配置好scp到其他节点即可（修改配置文件）**

**安装要求见**[**https://www.elastic.co/support/matrix**](https://www.elastic.co/support/matrix)

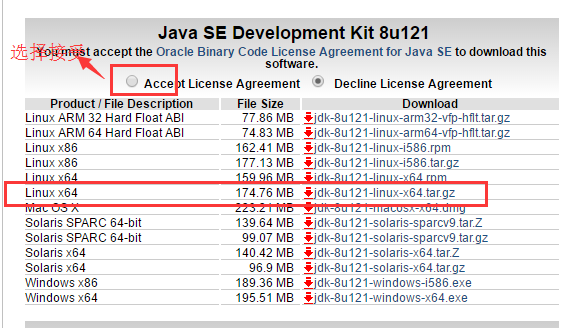
### 2.1.1下载

#### #redis下载

**wget** [**http://download.redis.io/releases/redis-3.0.5.tar.gz**](http://download.redis.io/releases/redis-3.0.5.tar.gz)

#### #jdk下载

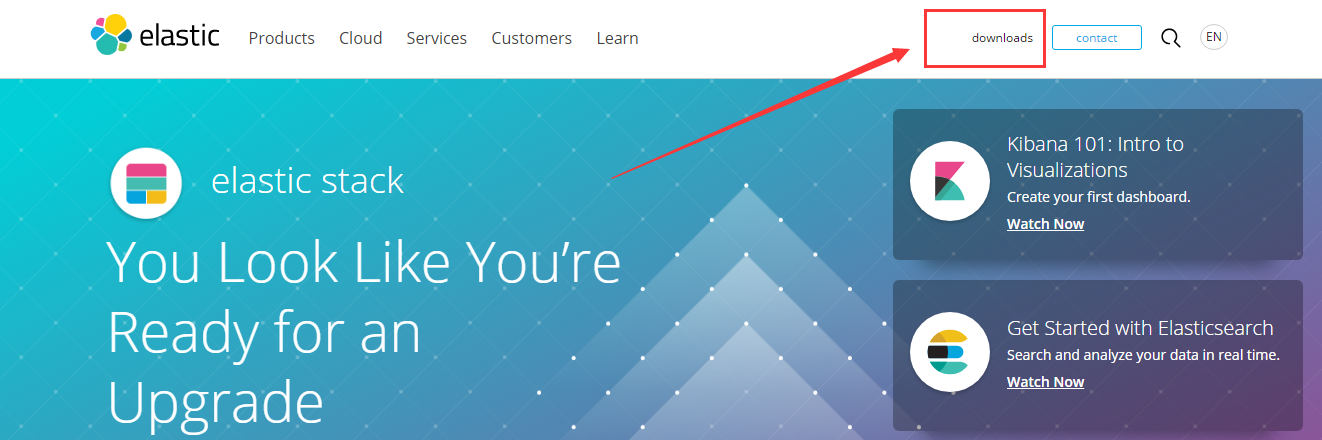
[**http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html**](http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html)



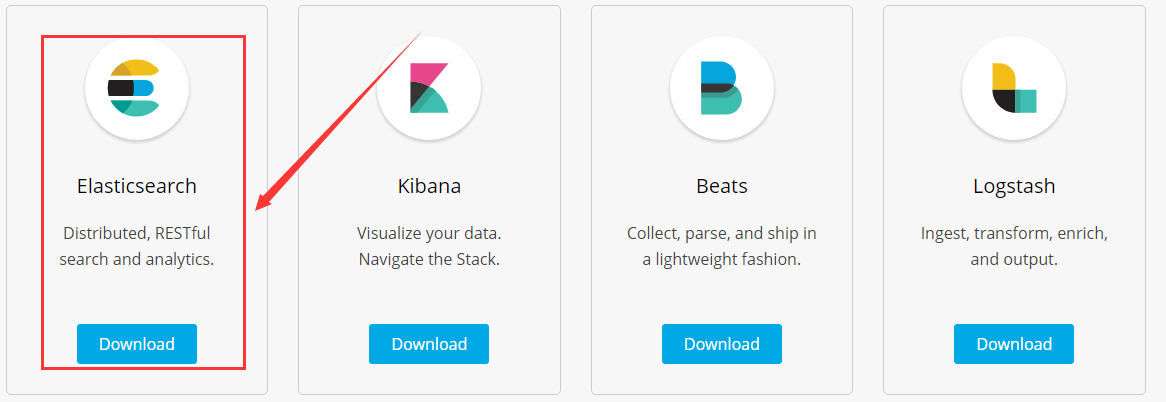
#### #Elasticsearch下载

**第一步，进入官网**[**https://www.elastic.co/**](https://www.elastic.co/)

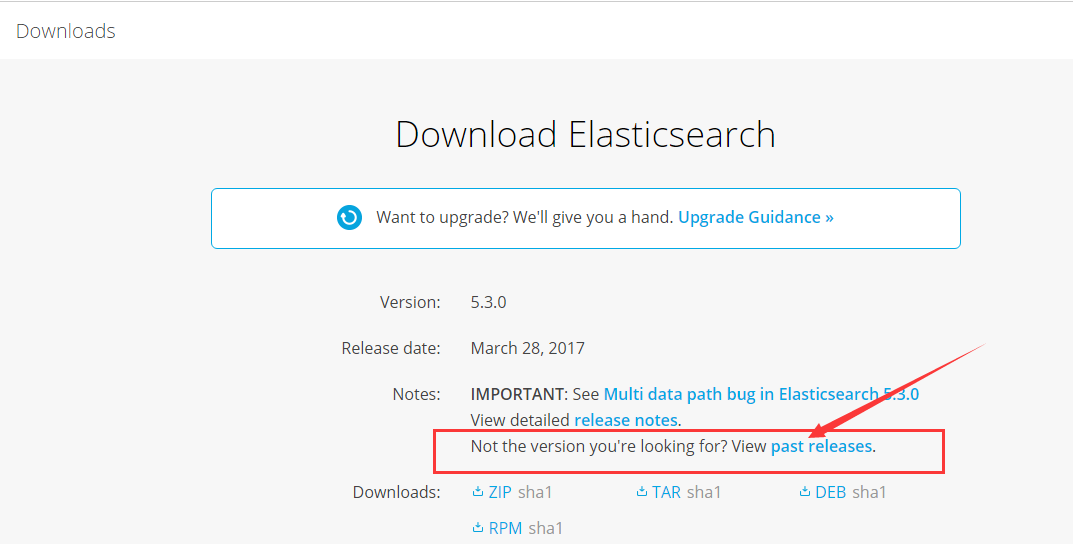
**点击下载**



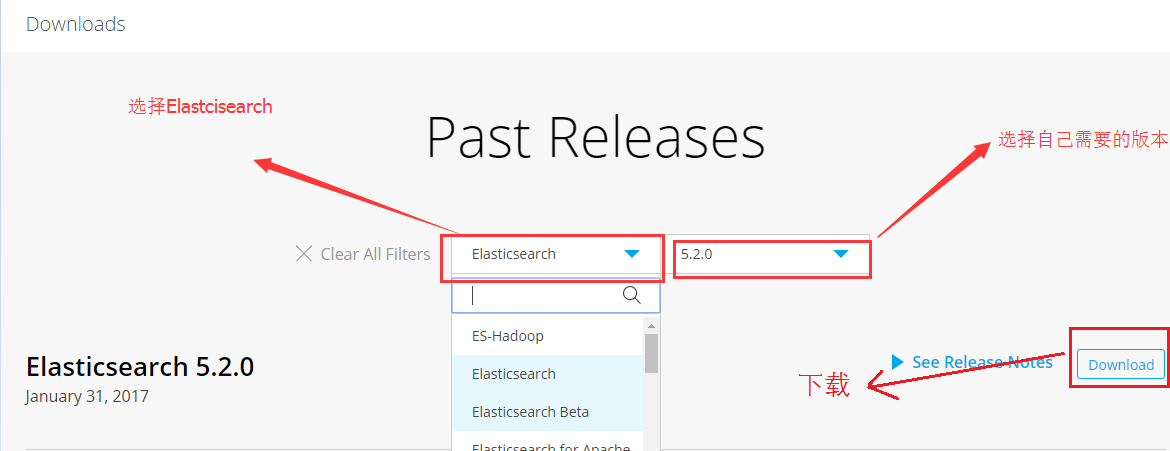
第二步点击elasticsearch download



**默认点击进去之后是最新版本，如下图**



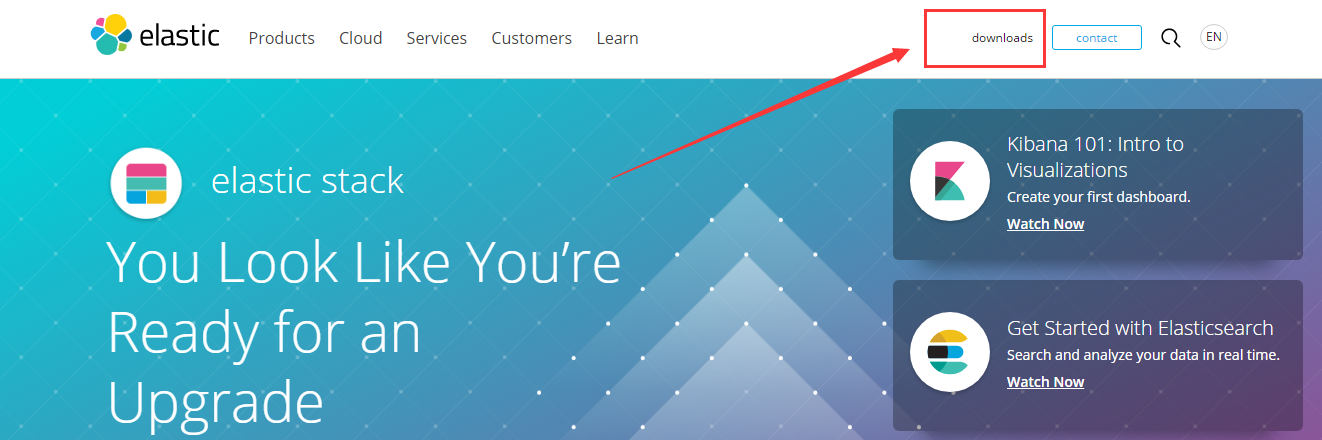
**第三点击红色部分，可以查看old版本**



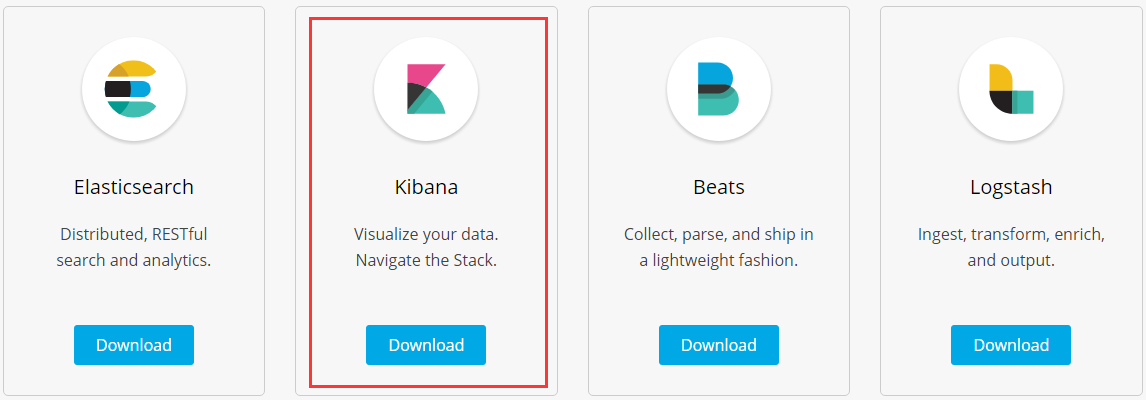
#### #下载Kibana

**第一步，进入官网**[**https://www.elastic.co/**](https://www.elastic.co/)

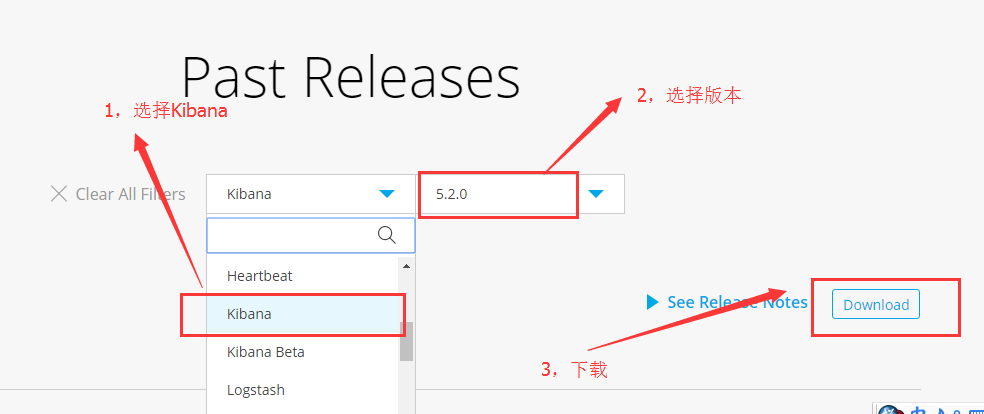
**点击下载**



第二步点击Kibana download



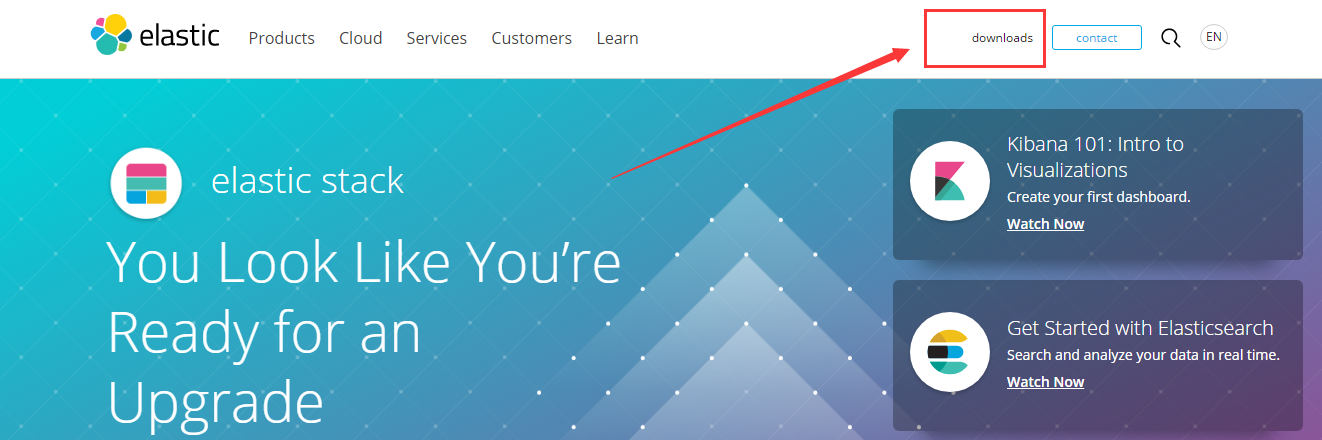
**第三步：点击past releases可以查看其它版本**



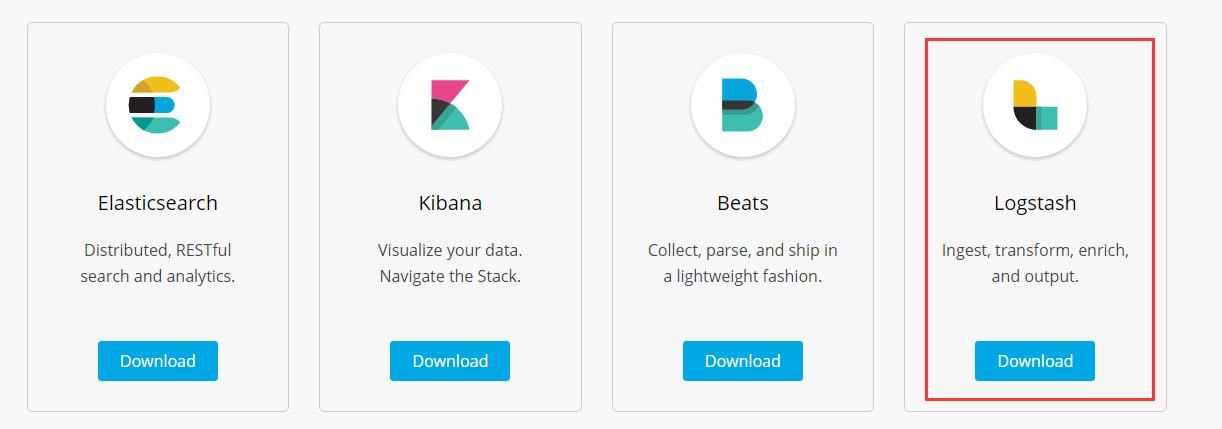
#### #下载Logstash

**第一步，进入官网**[**https://www.elastic.co/**](https://www.elastic.co/)

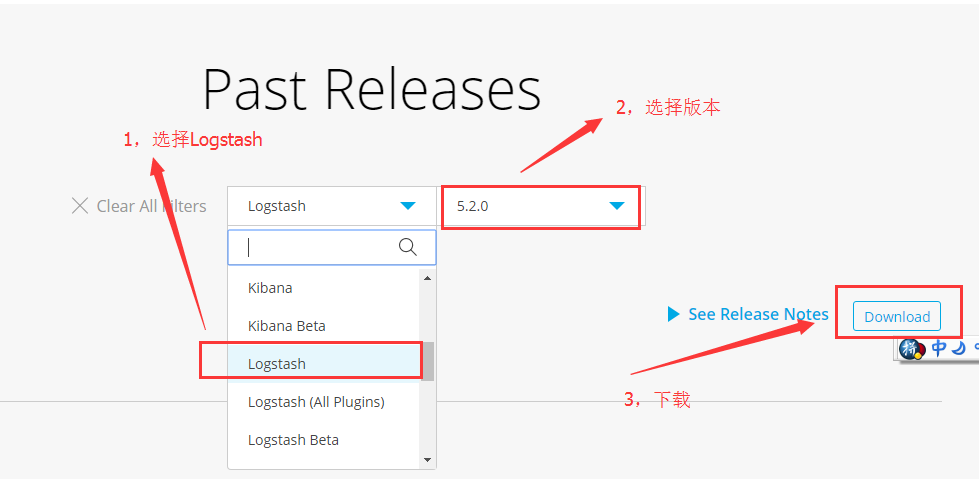
**点击下载**



第二步点击Logstash download



**第三步下载**



## 2.2 准备

### 2.2.1 关闭SELinux

**#下面的命令实现永久关闭SELinux**

**[root@elk ~]# sed -i 's/^SELINUX=.\*/#&/;s/^SELINUXTYPE=.\*/#&/;/SELINUX=.\*/a SELINUX=disabled' /etc/sysconfig/selinux**

**#永久修改下主机名，需要重启系统之后生效**

**#下面的命令实现临时关闭SELinux**

**[root@elk ~]# setenforce 0**

**setenforce: SELinux is disabled**

### 2.2.2 修改主机名

**#永久修改**

**[root@elk ~]#hostnamectl set-hostname elk.server.com**

**或者**

**[root@elk ~]# vi /etc/hostname**

**localhost.localdomain ------》把这行修改成下面的**

**elk.server.com #修改成你自己的主机名**

**#添加域名**

**[root@elk ~]#cat /etc/hosts**

**192.168.201.135 elk.server.com**

### 2.2.3 关闭firewall

**#临时关闭**

**[root@elk ~]# iptables -F**

**或者**

**[root@elk ～]# systemctl stop firewalld.service**

**#永久关闭，需要下次重启系统之后生效**

**[root@elk ~]# systemctl disable firewalld.service**

**Removed symlink /etc/systemd/system/dbus-org.fedoraproject.FirewallD1.service.**

**Removed symlink /etc/systemd/system/basic.target.wants/firewalld.service.**

### 2.2.4 同步时间

**#首先修改时区亚洲**

**#查看时区**

**[root@192 config]# ls -l /etc/localtime**

**lrwxrwxrwx. 1 root root 38 Apr 13 18:08 /etc/localtime -> ../usr/share/zoneinfo/America/New\_York**

**#删除link**

**[root@elk yum.repos.d]# ntpdate time.nist.gov**

**10 Apr 11:00:04 ntpdate[40122]: step time server 216.229.0.179 offset 53747.856066 sec**

### 2.2.5 修改open files

Why？

**启动elasticsearch出现如下错误**

**1，**

**ERROR: bootstrap checks failed**

**问题：最大文件描述符打开的太低，需要增加**

**max file descriptors [65535] for elasticsearch process likelytoo low, increase to at least [65536]**

**2，**

**问题：最大线程数，打开的太低，需要增加线程数**

**max number of threads [1024] for user [elasticsearch] likely toolow, increase to at least [2048]**

**解决：**

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

**\* soft nproc 2048**

**3，**

**问题：打开虚拟内存的个数太少需要增加**

**max virtual memory areas vm.max\_map\_count [65530] likely toolow, increase to at least [262144]**

**解决：**

**[root@elk ~]#vi /etc/sysctl.conf**

**vm.max\_map\_count=655360**

**[root@elk ~]#sysctl -p**

#### #查看打开文件数

**[root@elk ~]# ulimit -a**

**core file size (blocks, -c) 0**

**data seg size (kbytes, -d) unlimited**

**scheduling priority (-e) 0**

**file size (blocks, -f) unlimited**

**pending signals (-i) 18705**

**max locked memory (kbytes, -l) 64**

**max memory size (kbytes, -m) unlimited**

**open files (-n) 1024 -----------------》默认为1024**

**pipe size (512 bytes, -p) 8**

**POSIX message queues (bytes, -q) 819200**

**real-time priority (-r) 0**

**stack size (kbytes, -s) 8192**

**cpu time (seconds, -t) unlimited**

**max user processes (-u) 4096**

**virtual memory (kbytes, -v) unlimited**

**file locks (-x) unlimited**

#### #临时修改

**[root@elk ~]# ulimit -SHn 65536**

**注：**

**-S 设置软件资源限制**

**-H 设置硬件资源限制**

**-n 设置内核可以同时可以打开文件描述符**

**[root@elk ~]# ulimit -n**

**65536**

**注：**

**修改这个原因，启动elasticsearch 会出现这个情况too many open files,导致启动失败**

#### #永久修改

**#在文件最后添加**

**[root@elk ~]# vi /etc/security/limits.conf**

**\* soft nofile 65536**

**\* hard nofile 131072**

**\* soft nproc 2048**

**\* hard nproc 4096**

**注**：

**文件格式：username|@groupname type resource limit**

**分为3中类型type（有 soft，hard 和 -）**

**soft是指当前系统生效的设置值**

**hard 系统设置的最大值**

* **同时设置了soft和hard的值**

**nofile - 打开文件的最大数目**

**noproc - 进程的最大数目**

**soft<=hard soft的限制不能比hard限制高**

#需要重启系统才会生效

#### #修改kernel配置文件

**#在文件末尾添加**

**[root@elk ~]# vi /etc/sysctl.conf**

**vm.max\_map\_count=655360**

**#生效下**

**[root@elk ~]#sysctl -p**

**注：**

**vm.max\_map\_count文件允许max\_map\_count限制虚拟内存的数量**

**官方地址https://www.elastic.co/guide/en/elasticsearch/reference/current/vm-max-map-count.html**

## 2.3 安装

### 2.3.1 安装redis

#### #创建安装目录

**[root@elk yum.repos.d]#mkdir -pv /data/application/**

**[root@elk yum.repos.d]#wget http://download.redis.io/releases/redis-3.0.5.tar.gz**

#### #编译并进行安装

**[root@elk ~]# tar zxf redis-3.0.5.tar.gz && cd redis-3.0.5**

**[root@elk redis-3.0.5]# make PREFIX=/data/application/redis-3.0.5 install**

#### #创建配置文件目录

**[root@elk redis-3.0.5]#mkdir /data/application/redis-3.0.5/{etc,run,log}**

**#修改redis.conf**

**[root@elk redis-3.0.5]#vi /data/application/redis-3.0.5/etc/redis.conf**













**daemonize yes #后台模式运行**

**pidfile /data/application/redis-3.0.5/run/redis.pid** #redis的pid

**bind 192.168.201.135** #这里根据自己的ip填写

**port 6379 #端口**

**logfile "/data/application/redis-3.0.5/log/redis.log" #log存放位置**

**dir /data/application/redis-3.0.5**

#### #启动redis

**[root@elk~]#执行下面的命令**

**/data/application/redis-3.0.5/bin/redis-server /data/application/redis-3.0.5/etc/redis.conf**

**#查看是否启动成功**

**[root@elk ~]# lsof -i:6379**

**COMMAND PID USER FD TYPE DEVICE SIZE/OFF NODE NAME**

**redis-ser 2736 root 4u IPv4 26198 0t0 TCP 192.168.201.135:6379 (LISTEN)**

#### #测试redis

**[root@localhost ~]# /data/application/redis-3.0.5/bin/redis-cli -h 192.168.201.135**

**192.168.201.135:6379> ping**

PONG

**#出现PONG，证明可以使用**

### 2.3.2 安装jdk

#### #卸载系统自带jdk

**#查看系统自带jdk**

**[root@elk elk\_pack]#rpm -qa|grep java**

**#卸载**

**[root@elk elk\_pack]#yum remove java-1.8.0-openjdk**

**[root@elk elk\_pack]#yum remove java-1.7.0-openjdk**

#### #解压

**[root@elk elk\_pack]# tar zxvf jdk-8u91-linux-x64.tar.gz -C /data/application/**

**#注：**

**-C 指定解压到指定的路径**

#### #配置jdk环境变量

**#全局配置**

**[root@elk jdk1.8.0\_91]# vi /etc/profile.d/java.sh**

**export JAVA\_HOME=/data/application/jdk1.8.0\_91/**

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

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

**[root@elk jdk1.8.0\_91]#source /etc/profile.d/java.sh**

**#查看版本**

**[root@elk jdk1.8.0\_91]#java -version**

**java version "1.8.0\_91"#出现这个证明成功**

**Java(TM) SE Runtime Environment (build 1.8.0\_91-b14)**

**Java HotSpot(TM) 64-Bit Server VM (build 25.91-b14, mixed mode)**

### 2.4.2 安装elasticsearch

#### #创建elk用户

**[root@localhost application]# adduser -s /bin/bash -c 'elk' -m -d /home/elk elk**

**注：**

**从2.0开始不能用root用户启动需要elk用户启动**

#### #解压

**[root@elk elk\_pack]# tar zxvf elasticsearch-5.2.0.tar.gz -C /data/application/**

**注：**

**Elasticsearch是不需要编译的，解压就可以使用**

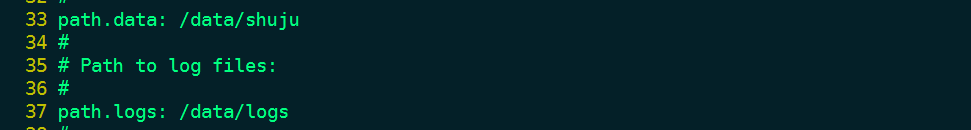
#### #修改配置文件

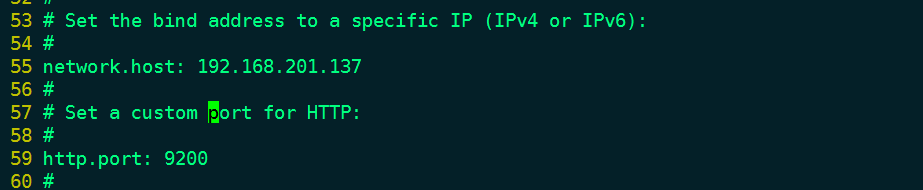
**#备份配置文件**

**[root@elk ~]# cp /data/application/elasticsearch-5.2.0/config/elasticsearch.yml{,.ori}**

**#找到以下几行修改**

**[root@elk config]# vi elasticsearch.yml**





**path.data: /data/shuju ----》存放数据路径**

**path.logs: /data/logs -----》日志路径**

**network.host: 192.168.201.137 -----》根据自己的ip修改**

**http.port: 9200**

**#创建los,shuju**

**[root@elk config]#mkdir /data/{shuju,logs}**

#### #启动elasticsearch

**#修改elasticsearch权限**

**[root@elk ~]#chown -R elk.elk /data/application/elasticsearch-5.2.0/**

**[root@elk ~]#chown -R elk.elk /data/{shuju,logs}**

**[root@elk ~]# su – elk**

**#在前台显示下效果**

**[elk@elk ~]$/data/application/elasticsearch-5.2.0/bin/elasticsearch**

**[2017-04-12T22:29:46,478][INFO ][o.e.n.Node ] [] initializing ...**

**[2017-04-12T22:29:46,714][INFO ][o.e.e.NodeEnvironment ] [z8htm2J] using [1] data paths, mounts [[/ (rootfs)]], net usable\_space [8gb], net total\_space [16.9gb], spins? [unknown], types [rootfs]**

**[2017-04-12T22:29:46,714][INFO ][o.e.e.NodeEnvironment ] [z8htm2J] heap size [1.9gb], compressed ordinary object pointers [true]**

**[2017-04-12T22:29:46,755][INFO ][o.e.n.Node ] node name [z8htm2J] derived from node ID [z8htm2JoRRaW0bsTgx-rhQ]; set [node.name] to override**

**[2017-04-12T22:29:46,756][INFO ][o.e.n.Node ] version[5.3.0], pid[4012], build[3adb13b/2017-03-23T03:31:50.652Z], OS[Linux/3.10.0-514.10.2.el7.x86\_64/amd64],**

**mustache]**

**中间内容省略。。。。。**

**[2017-04-12T22:29:51,352][INFO ][o.e.p.PluginsService ] [z8htm2J] no plugins loaded**

**into cluster\_state**

**[2017-04-12T22:30:04,658][INFO ][o.e.c.r.a.AllocationService] [z8htm2J] Cluster health status changed from [RED] to [YELLOW]**

**注：**

**如果执行完这个脚本，中途没有退出，说明启动成功**

#### #测试是否成功

**[root@elk ~]# curl 192.168.201.135:9200**

**{**

**"name" : "z8htm2J",**

**"cluster\_name" : "elasticsearch",**

**"cluster\_uuid" : "wEbF7BwgSe-0vFyHb1titQ",**

**"version" : {**

**"number" : "5.3.0",**

**"build\_hash" : "3adb13b",**

**"build\_date" : "2017-03-23T03:31:50.652Z",**

**"build\_snapshot" : false,**

**"lucene\_version" : "6.4.1"**

**},**

**"tagline" : "You Know, for Search"**

**}**

**出现以上情况，证明elasticsearch可以正常使用**

### 2.4.3 安装Logstash

#### #解压

**[root@elk elk\_pack]# tar zxvf logstash-5.2.0.tar.gz -C /data/application/**

**注：**

**Logstash是不需要编译的，解压就可以使用**

#### #测试能否使用

**[root@elk ~]# /data/application/logstash-5.2.0/bin/logstash -e 'input { stdin { } } output {stdout {} }'**

**Sending Logstash's logs to /data/application/logstash-5.2.0/logs which is now configured via log4j2.properties**

**The stdin plugin is now waiting for input:**

**[2017-04-12T11:54:10,457][INFO ][logstash.pipeline ] Starting pipeline {"id"=>"main", "pipeline.workers"=>1, "pipeline.batch.size"=>125, "pipeline.batch.delay"=>5, "pipeline.max\_inflight"=>125}**

**[2017-04-12T11:54:10,481][INFO ][logstash.pipeline ] Pipeline main started**

**[2017-04-12T11:54:10,563][INFO ][logstash.agent ] Successfully started Logstash API endpoint {:port=>9600}**

**hello world ---->输入hell world，随便输入什么，能输出就证明可以使用**

**2017-04-12T03:54:40.278Z localhost.localdomain hello world ---->输出hello world**

**/data/application/logstash-5.2.0/vendor/bundle/jruby/1.9/gems/logstash-patterns-core-4.0.2/patterns**

**#证明可以使用**

### 2.4.4 安装Kibana

#### #解压

[**root@elk elk\_pack]# tar zxvf kibana-5.2.0-linux-x86\_64.tar.gz -C /data/application/**

**注：**

**Kibana是不需要编译的，解压就可以使用**

#### #修改配置kibana.yml文件

**#cd kibana这个目录**

**[root@elk ~]# cd /data/application/kibana-5.2.0/config/**

**#找到以下几行修改**

**[root@elk config]# egrep -v "^$|^[#]" kibana.yml**

**server.port: 5601 #kibana的端口**

**server.host: "192.168.201.135" #访问kibana的ip地址**

**elasticsearch.url: "http://192.168.201.135:9200" #elasticsearch的ip地址**

**kibana.index: ".kibana" #创建索引**

#### #测试是否启动成功

**[root@192 ~]# /data/application/kibana-5.2.0/bin/kibana**

**log [06:22:02.940] [info][status][plugin:kibana@5.2.0] Status changed from uninitialized to green - Ready**

**log [06:22:03.106] [info][status][plugin:elasticsearch@5.2.0] Status changed from uninitialized to yellow - Waiting for Elasticsearch**

**log [06:22:03.145] [info][status][plugin:console@5.2.0] Status changed from uninitialized to green - Ready**

**log [06:22:03.193] [warning] You're running Kibana 5.2.0 with some different versions of Elasticsearch. Update Kibana or Elasticsearch to the same version to prevent compatibility issues: v5.3.0 @ 192.168.201.135:9200 (192.168.201.135)**

**log [06:22:05.728] [info][status][plugin:timelion@5.2.0] Status changed from uninitialized to green - Ready**

**log [06:22:05.744] [info][listening] Server running at http://192.168.201.135:5601**

**log [06:22:05.746] [info][status][ui settings] Status changed from uninitialized to yellow - Elasticsearch plugin is yellow**

**log [06:22:08.263] [info][status][plugin:elasticsearch@5.2.0] Status changed from yellow to yellow - No existing Kibana index found**

**log [06:22:09.446] [info][status][plugin:elasticsearch@5.2.0] Status changed from yellow to green - Kibana index ready**

**log [06:22:09.447] [info][status][ui settings] Status changed from yellow to green – Ready**

**#证明启动成功**

#### #查看port

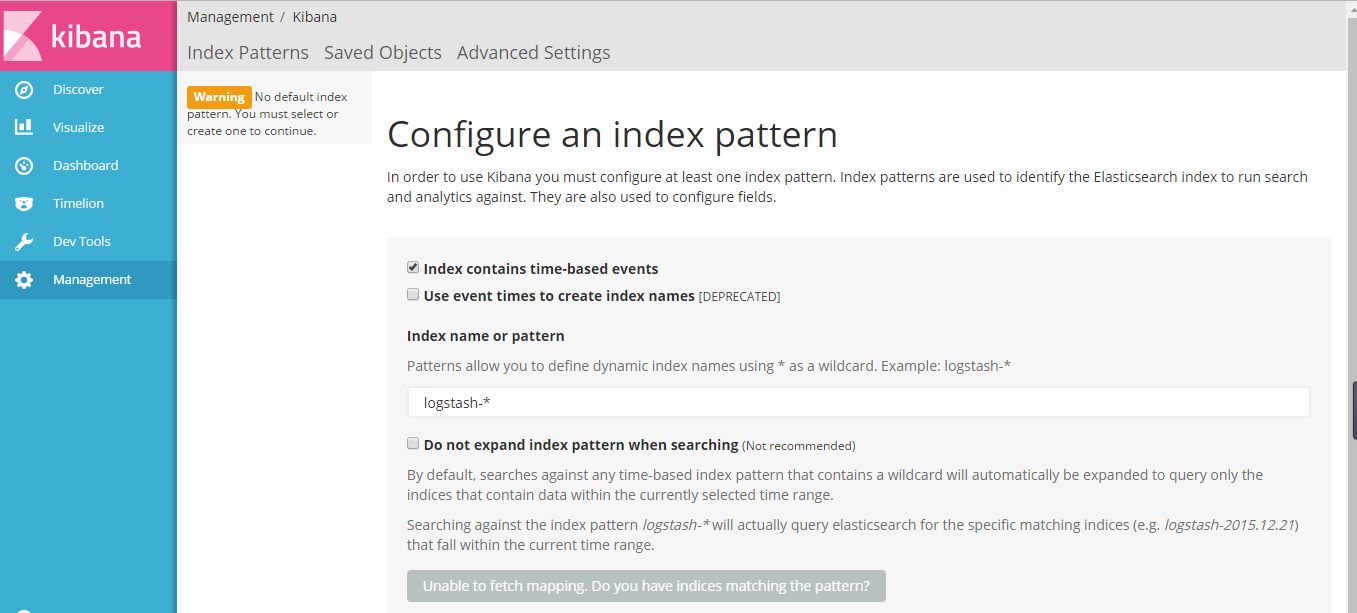
**[root@elk shuju]# lsof -i:5601**

**COMMAND PID USER FD TYPE DEVICE SIZE/OFF NODE NAME**

**node 4690 root 13u IPv4 40663 0t0 TCP 192.168.201.135:esmagent (LISTEN)**

#### #通过web访问

**http://192.168.201.135:5601**



**注：**

**访问成功，证明Kibana可以使用，由于第一次使用，这个时候还不能使用，一定要注意这一点，只有数据收集好时，才可以使用（这里是一个小小的坑，否则对于新手来说，以为是不能使用）**

## 2.4 实战

### 2.4.1 Nginx访问日志

以下是在Agent端配置

##### #安装Logstash

**这里安装logstash和在服务端安装logstash方式一样，区别在于配置文件有区别，这里是采集数据**

**请参考服务端的安装方式**

##### #创建agent目录

**#方便管理文件，把采集数据的文件统一放到一个目录中管理**

**[root@localhost logstash-5.2.0]# mkdir agent\_conf**

**#cd to agent\_conf**

**[root@localhost logstash-5.2.0]# cd agent\_conf/**

##### #创建采集数据的文件

**[root@localhost agent\_conf]# cat nginx-agent.conf**

**input {**

**file {**

**path => ["/var/log/nginx/hxopensource-access.log"]**

**type => "nginx\_access"**

**}**

**}**

**output {**

**stdout {codec => rubydebug }**

**if [type] == "nginx\_access"{**

**redis {**

**host => ["192.168.201.135:6379"]**

**data\_type=> "list"**

**key => "nginx"**

**}**

**}**

**}**

##### #测试ngnix

**[root@localhost conf]# curl localhost**

**hello ansible install nginx**

##### #配置nginx日志格式

**#编辑vi nginx.conf**

**20 log\_format backend '$remote\_addr - $remote\_user[$time\_local] "$request" '**

**21 '$status $body\_bytes\_sent "$http\_referer" '**

**22 '"$http\_user\_agent" "$http\_x\_forwarded\_for" $request\_time ';**

##### #查看访问日志

**[root@localhost conf]# cat /var/log/nginx/hxopensource-access.log**

**192.168.201.1 - -[24/Mar/2017:04:24:18 +0800] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/57.0.2987.133 Safari/537.36" "-" 0.000**

**至此agent端配置完成**

##### #Logstash-server端配置

**#方便管理文件，把采集数据的文件统一放到一个目录中管理**

**#在server端执行**

**[root@elk logstash-5.2.0]# mkdir server\_conf**

**#cd to server\_conf**

**[root@localhost logstash-5.2.0]# cd server\_conf/**

**#创建接受文件**

**[root@elk server\_conf]#cat logstash-server-nginx.conf**

**input {**

**redis {**

**host => "192.168.201.135"**

**port => 6379**

**data\_type => "list"**

**key => "nginx"**

**}**

**}**

**filter {**

**if[type] =~ "NginxAccess-" {**

**grok {**

**patterns\_dir => ["/data/application/logstash-5.2.0/vendor/bundle/jruby/1.9/gems/logstash-patterns-core-4.0.2/patterns/"]**

**match => {**

**"message" => "%{NGINXACCESS}"**

**}**

**}**

**date {**

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

**}**

**}**

**}**

**output {**

**elasticsearch {**

**action => "index"**

**hosts => "192.168.201.135:9200"**

**index => "nginx-%{+yyyy.MM.dd}"**

**}**

**}**

##### #测试

**启动顺序由左到右Elasticsearch-🡪Kibana--🡪Logstash**

**#在server端启动Elasticsearch、Kibana、Logstashserver端**

**#以普通用户启动Elasticsearch**

**[elk@localhost ~]$ /data/application/elasticsearch-5.2.0/bin/elasticsearch**

**[2017-04-13T14:47:07,502][INFO ][o.e.n.Node ] [] initializing ...**

**[2017-04-13T14:47:07,631][INFO ][o.e.e.NodeEnvironment ] [ihv1bLE] using [1] data paths, mounts [[/ (rootfs)]], net usable\_space [8gb], net total\_space [16.9gb], spins? [unknown], types [rootfs]**

**中间部分启动内容省略。。。**

**[2017-04-13T14:47:19,122][INFO ][o.e.c.r.a.AllocationService] [ihv1bLE] Cluster health status changed from [RED] to [YELLOW] (reason: [shards started [[.kibana][0]] ...]).**

#在server端启动Kibana

**[root@localhost ~]# /data/application/kibana-5.2.0-linux-x86\_64/bin/kibana**

**kibana kibana-plugin**

**[root@localhost ~]# /data/application/kibana-5.2.0-linux-x86\_64/bin/kibana**

**log [06:48:15.408] [info][status][plugin:kibana@5.2.0] Status changed from uninitialized to green - Ready**

**log [06:48:15.519] [info][status][plugin:elasticsearch@5.2.0] Status changed from uninitialized to yellow - Waiting for Elasticsearch**

**log [06:48:15.573] [info][status][plugin:console@5.2.0] Status changed from uninitialized to green - Ready**

**log [06:48:16.043] [info][status][plugin:timelion@5.2.0] Status changed from uninitialized to green - Ready**

**log [06:48:16.048] [info][listening] Server running at http://192.168.201.135:5601**

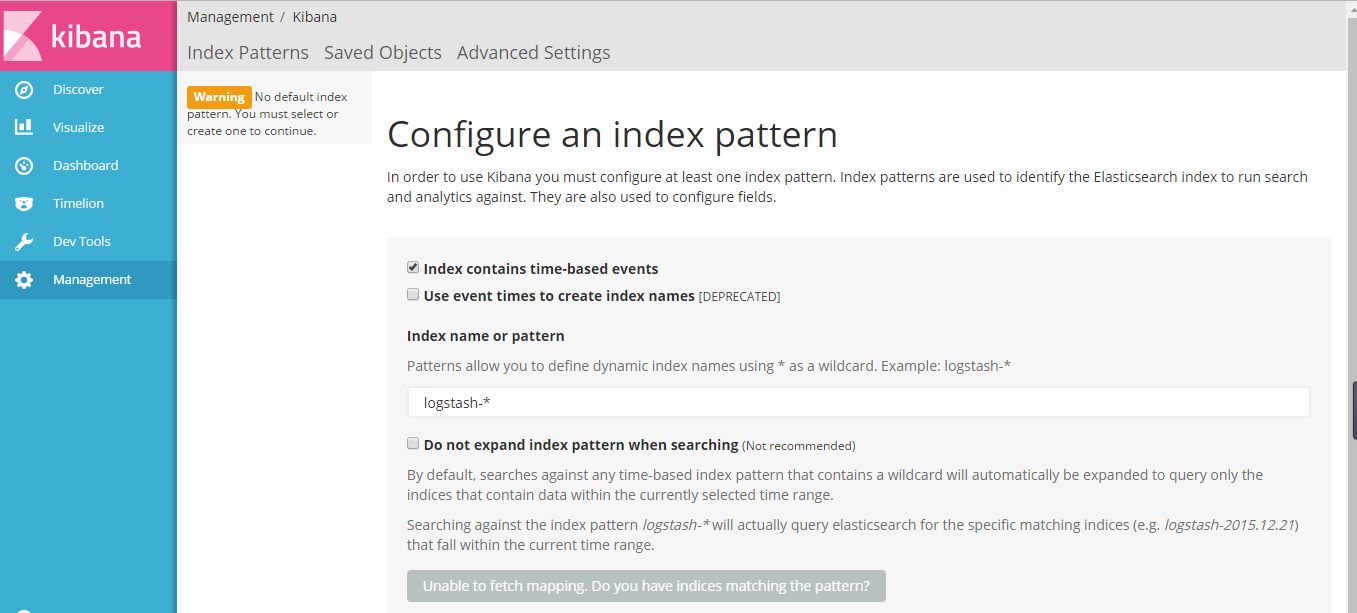
**log [06:48:16.050] [info][status][ui settings] Status changed from uninitialized to yellow - Elasticsearch plugin is yellow**

**log [06:48:16.262] [info][status][plugin:elasticsearch@5.2.0] Status changed from yellow to green - Kibana index ready**

**log [06:48:16.263] [info][status][ui settings] Status changed from yellow to green – Ready**

**#访问下Kibana**

**192.168.201.135:5601**



#在server端启动logstash

#测试语法是否有问题

**[root@localhost logstash-5.2.0]# ./bin/logstash -f logstash-nginx.conf -t**

**Sending Logstash's logs to /data/application/logstash-5.2.0/logs which is now configured via log4j2.properties**

**Configuration OK**

**[2017-04-13T14:58:15,860][INFO ][logstash.runner ] Using config.test\_and\_exit mode. Config Validation Result: OK. Exiting Logstash**

**#启动**

**[root@localhost logstash-5.2.0]# ./bin/logstash -f logstash-nginx.conf**

**Sending Logstash's logs to /data/application/logstash-5.2.0/logs which is now configured via log4j2.properties**

**[2017-04-13T14:58:51,006][INFO ][logstash.inputs.redis ] Registering Redis {:identity=>"redis://@192.168.201.135:6379/0 list:nginx"}**

**[2017-04-13T14:58:51,917][INFO ][logstash.outputs.elasticsearch] Elasticsearch pool URLs updated {:changes=>{:removed=>[], :added=>[http://192.168.201.135:9200/]}}**

**[2017-04-13T14:58:51,918][INFO ][logstash.outputs.elasticsearch] Running health check to see if an Elasticsearch connection is working {:healthcheck\_url=>http://192.168.201.135:9200/, :path=>"/"}**

**[2017-04-13T14:58:52,292][WARN ][logstash.outputs.elasticsearch] Restored connection to ES instance {:url=>#<URI::HTTP:0x68cdb128 URL:http://192.168.201.135:9200/>}**

**[2017-04-13T14:58:52,297][INFO ][logstash.outputs.elasticsearch] Using mapping template from {:path=>nil}**

**[2017-04-13T14:58:52,693][INFO ][logstash.outputs.elasticsearch] Attempting to install template {:manage\_template=>{"template"=>"logstash-\*", "version"=>50001, "settings"=>{"index.refresh\_interval"=>"5s"}, "mappings"=>{"\_default\_"=>{"\_all"=>{"enabled"=>true, "norms"=>false}, "dynamic\_templates"=>[{"message\_field"=>{"path\_match"=>"message", "match\_mapping\_type"=>"string", "mapping"=>{"type"=>"text", "norms"=>false}}}, {"string\_fields"=>{"match"=>"\*", "match\_mapping\_type"=>"string", "mapping"=>{"type"=>"text", "norms"=>false, "fields"=>{"keyword"=>{"type"=>"keyword"}}}}}], "properties"=>{"@timestamp"=>{"type"=>"date", "include\_in\_all"=>false}, "@version"=>{"type"=>"keyword", "include\_in\_all"=>false}, "geoip"=>{"dynamic"=>true, "properties"=>{"ip"=>{"type"=>"ip"}, "location"=>{"type"=>"geo\_point"}, "latitude"=>{"type"=>"half\_float"}, "longitude"=>{"type"=>"half\_float"}}}}}}}}**

**[2017-04-13T14:58:52,713][INFO ][logstash.outputs.elasticsearch] Installing elasticsearch template to \_template/logstash**

**[2017-04-13T14:58:53,037][INFO ][logstash.outputs.elasticsearch] New Elasticsearch output {:class=>"LogStash::Outputs::ElasticSearch", :hosts=>[#<URI::Generic:0x473099d9 URL://192.168.201.135:9200>]}**

**[2017-04-13T14:58:53,512][INFO ][logstash.pipeline ] Starting pipeline {"id"=>"main", "pipeline.workers"=>1, "pipeline.batch.size"=>125, "pipeline.batch.delay"=>5, "pipeline.max\_inflight"=>125}**

**[2017-04-13T14:58:53,526][INFO ][logstash.pipeline ] Pipeline main started**

**[2017-04-13T14:58:53,614][INFO ][logstash.agent ] Successfully started Logstash API endpoint {:port=>9600}**

**#在agent端启动logstash**

**[root@localhost logstash-5.2.0]# ./bin/logstash -f logstash-nginx.conf**

**Sending Logstash's logs to /root/logstash-5.2.0/logs which is now configured via log4j2.properties**

**[2017-03-24T03:53:45,873][INFO ][logstash.pipeline ] Starting pipeline {"id"=>"main", "pipeline.workers"=>1, "pipeline.batch.size"=>125, "pipeline.batch.delay"=>5, "pipeline.max\_inflight"=>125}**

**[2017-03-24T03:53:45,890][INFO ][logstash.pipeline ] Pipeline main started**

**[2017-03-24T03:53:45,976][INFO ][logstash.agent ] Successfully started Logstash API endpoint {:port=>9600}**

**至此所有服务启动完成**

**#查看redis中的数据**

**[root@client~]#/data/application/redis-3.0.5/bin/redis-cli -h 192.168.201.137**

**192.168.201.137:6379> keys \***

**1) "nginx\_access"**

**注意：**

**由于这里还没有使用logstash从redis中取出数据，所以这里是可以看到消息队列中的数据，如果是生产环境，可能使用KEYS \*看不到消息队列中有数据，说明logstash已经从消息队列中取走了数据.**

**#在agent执行，，让nginx产生数据源**

**[root@localhost ~]# curl localhost**

**hello ansible install nginx**

**#查看下agent端采集的数据**

**[root@localhost logstash-5.2.0]# ./bin/logstash -f logstash-nginx.conf**

**Sending Logstash's logs to /root/logstash-5.2.0/logs which is now configured via log4j2.properties**

**[2017-03-24T03:53:45,873][INFO ][logstash.pipeline ] Starting pipeline {"id"=>"main", "pipeline.workers"=>1, "pipeline.batch.size"=>125, "pipeline.batch.delay"=>5, "pipeline.max\_inflight"=>125}**

**[2017-03-24T03:53:45,890][INFO ][logstash.pipeline ] Pipeline main started**

**[2017-03-24T03:53:45,976][INFO ][logstash.agent ] Successfully started Logstash API endpoint {:port=>9600}**

**{**

**"path" => "/var/log/nginx/hxopensource-access.log",**

**"@timestamp" => 2017-03-23T19:53:53.877Z,**

**"@version" => "1",**

**"host" => "localhost.localdomain",**

**"message" => "127.0.0.1 - -[24/Mar/2017:03:53:53 +0800] \"GET / HTTP/1.1\" 200 28\"-\" \"curl/7.19.7 (x86\_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.15.3 zlib/1.2.3 libidn/1.18 libssh2/1.4.2\"\"-\" 0.000 ",**

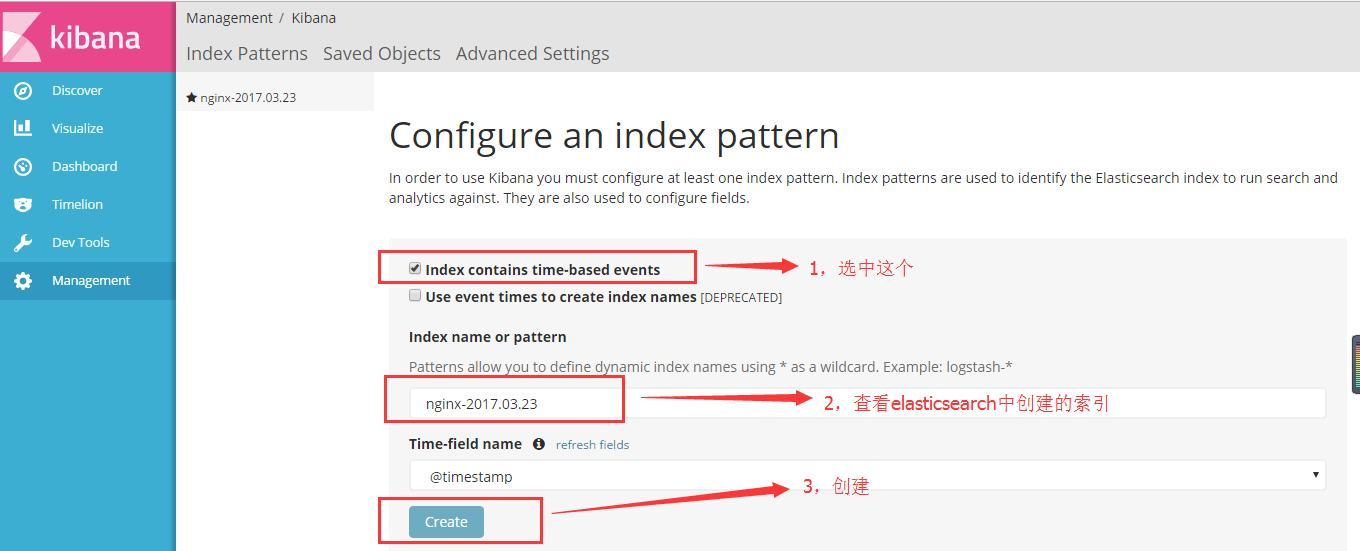
**"type" => "nginx\_access"**

**}**

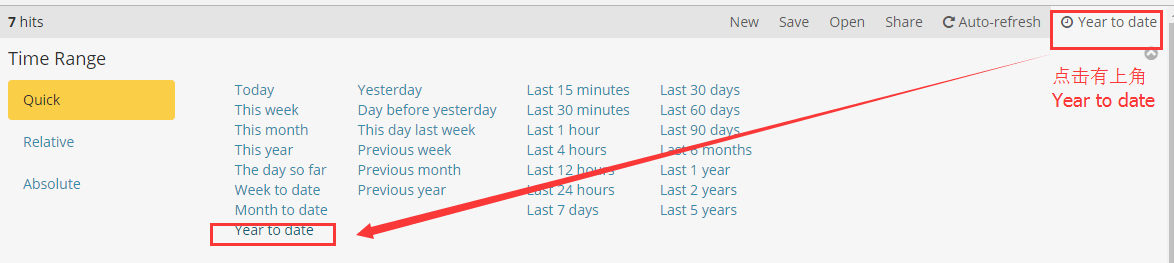
**黄色部分为采集到nginx访问日志**

##### 在Kibana上查看

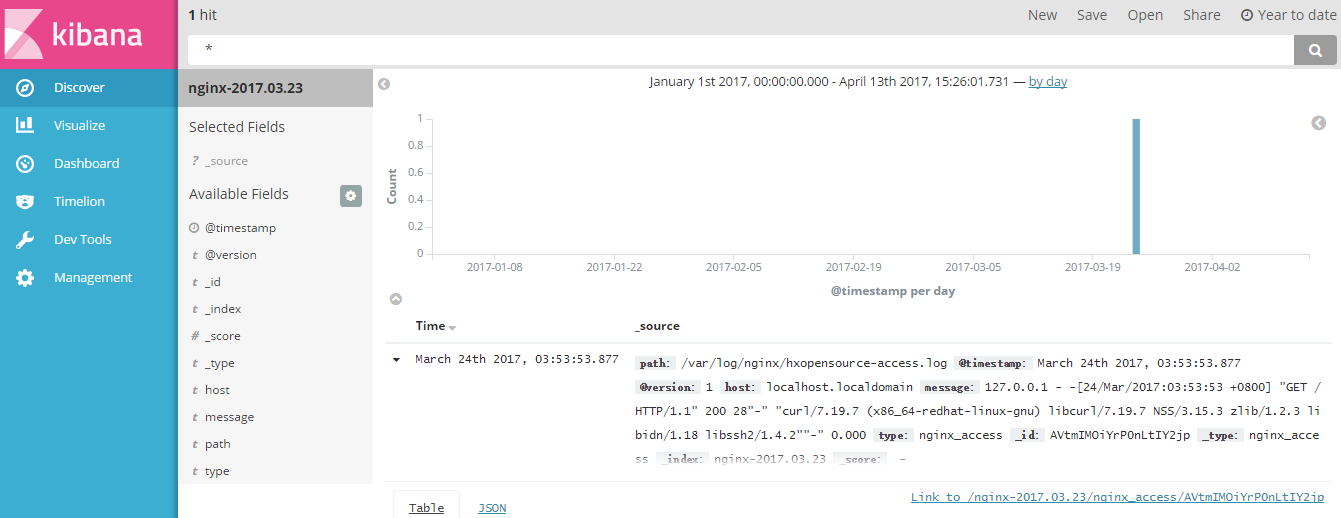
**#创建index**



#查看



#查看nginx访问日志



# FAQ

ERROR: bootstrap checks failed

问题：max file descriptors [65535] for elasticsearch process likelytoo low, increase to at least [65536]

解决：vi /etc/security/limits.conf

\*             -       nofile          65536

或者

\* soft nofile 65536

\* hard nofile 131072

\* soft nproc 2048

\* hard nproc 4096

问题：max number of threads [1024] for user [elasticsearch] likely toolow, increase to at least [2048]

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

\* soft nproc 2048

需重启生效。

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

解决：vi /etc/sysctl.conf

vm.max\_map\_count=655360

sysctl -p

## 官方文档

官方文档

<https://www.elastic.co/guide/index.html>