部署ELK分布式日志分析平台

**一、ELK应用背景：**

****

**需求背景：**生产环境中业务发展越来越庞大，服务器越来越多，各种访问日志、应用日志、错误日志量越来越多，导致运维人员无法很好的去管理日志，开发人员排查问题，需要到服务器上查日志，不方便，运营人员需要一些数据，需要我们运维到服务器上分析日志；

**解决的问题：**一般我们需要进行日志分析场景：直接在日志文件中 grep、awk 就可以获得自己想要的信息。但在规模较大也就是日志量多而复杂的场景中，此方法效率低下，面临问题包括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。需要集中化的日志管理，所有服务器上的日志收集汇总。常见解决思路是建立集中式日志收集系统，将所有节点上的日志统一收集，管理，访问；

**日志系统：**大型系统通常都是一个分布式部署的架构，不同的服务模块部署在不同的服务器上，问题出现时，大部分情况需要根据问题暴露的关键信息，定位到具体的服务器和服务模块，构建一套集中式日志系统，可以提高定位问题的效率；

一个完整的集中式日志系统，需要包含以下几个主要特点：

1.收集－能够采集多种来源的日志数据；

2.传输－能够稳定的把日志数据传输到中央系统；

3.存储－如何存储日志数据；

4.分析－可以支持 UI 分析；

5.警告－能够提供错误报告，监控机制；

**二、ELK平台概述：**

**概述：**Elasticsearch是个开源分布式搜索引擎，提供搜集、分析、存储数据三大功能；ELK是三个开源软件的缩写，分别为：Elasticsearch 、 Logstash以及Kibana , 它们都是开源软件。不过现在还新增了一个Beats，它是一个轻量级的日志收集处理工具（Agent），Beats占用资源少，适合于在各个服务器上搜集日志后传输给Logstash，官方也推荐此工具，目前由于原本的ELK Stack成员中加入了 Beats 工具所以已改名为Elastic Stack。而ELK则提供了一整套解决方案，并且都是开源软件，之间互相配合使用，完美衔接，高效的满足了很多场合的应用；

**特点：**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。详细可参考[Elasticsearch权威指南](https://www.elastic.co/guide/cn/elasticsearch/guide/current/index.html)；

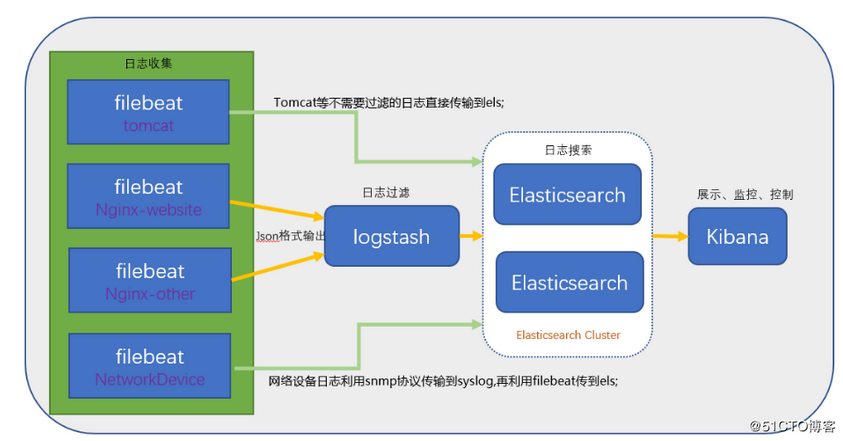
**网站：**

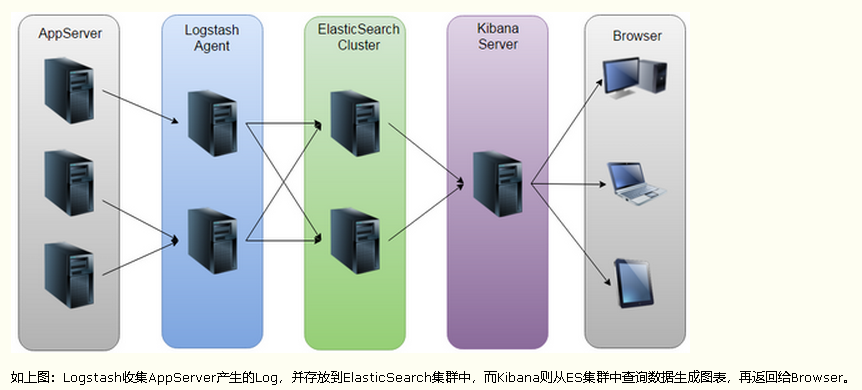
ELK官网：<https://www.elastic.co/cn/>

中文指南：<https://www.gitbook.com/book/chenryn/elk-stack-guide-cn/details>

**三、ELK核心组件：**

**ELK核心架构：**

****



**ELK核心组件：**

* Logstash：通过对日志进行收集、过滤、分析，支持大量的数据获取方法，并将其存储供以后使用（如搜索）。说到搜索，logstash带有一个web界面，搜索和展示所有日志。一般工作方式为c/s架构，client端安装在需要收集日志的主机上，server端负责将收到的各节点日志进行过滤、修改等操作在一并发往elasticsearch上去；
* ElasticSearch：是一个基于Lucene的开源分布式搜索服务器。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。在elasticsearch中，所有节点的数据是均等的；
* Kibana ：是一个基于浏览器页面的Elasticsearch前端展示工具，也是一个开源和免费的工具，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；
* Filebeat：隶属于Beats。目前Beats包含四种工具：

Packetbeat（搜集网络流量数据）；

Topbeat（搜集系统、进程和文件系统级别的 CPU 和内存使用情况等数据）；

Filebeat（搜集文件数据）；

Winlogbeat（搜集 Windows 事件日志数据）；

注：ELK Stack （5.0版本之后）--> Elastic Stack == （ELK Stack + Beats）

目前Beats包含六种工具：

Packetbeat： 网络数据（收集网络流量数据）

Metricbeat： 指标 （收集系统、进程和文件系统级别的 CPU 和内存使用情况等数据）

Filebeat： 日志文件（收集文件数据）

Winlogbeat： windows事件日志（收集 Windows 事件日志数据）

Auditbeat：审计数据 （收集审计日志）

Heartbeat：运行时间监控 （收集系统运行时的数据）

x-pack工具：x-pack对Elastic Stack提供了安全、警报、监控、报表、图表于一身的扩展包，是收费的，所以本文不涉及x-pack的安装；

**四、案例：部署ELK分布式日志分析平台：**

**案例拓扑：**

**案例环境：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 系统类型 | IP地址 | 主机名 | 所需软件 | 内存 |
| Centos 7.4 1708 64bit | 192.168.100.108 | master-node | java-1.8.0-openjdk  elasticsearch  kibana | 2G |
| Centos 7.4 1708 64bit | 192.168.100.107 | data-node1 | java-1.8.0-openjdk  elasticsearch  Logstash | 2G |
| Centos 7.4 1708 64bit | 192.168.100.106 | data-node2 | java-1.8.0-openjdk  elasticsearch | 2G |

**案例步骤：**

* 配置基础环境域名解析（在此只展示master-node节点的配置）；
* 所有节点安装JDK1.8（在此只展示master-node节点的配置）；
* 所有节点安装elasticsearch（在此只展示master-node节点的配置）；
* 配置es的master主节点；
* 配置es的data1节点：
* 配置es的data2节点：
* 配置es主节点开启es服务：
* 配置es从节点启动es服务：
* 在es的master主节点验证集群状态：
* 在es集群的master主节点安装并配置启动kibana：
* 使用web浏览器访问kibana的web页面进行验证：
* 在es集群的data1从节点安装并测试配置启动logstash；
* 配置logstash服务让收集的日志信息输出到es服务器中，而不是当前终端：
* 访问kibana服务器查看日志信息：
* 配置kibana的web网页设置索引：
* 配置kibana的web页面显示日志收集信息：
* 总结流程：
* **配置基础环境域名解析（在此只展示master-node节点的配置）；**

[root@master-node ~]# cat <<END >>/etc/hosts

192.168.100.108 master-node

192.168.100.107 data-node1

192.168.100.106 data-node2

END

[root@master-node ~]# yum -y install ntpdate

[root@master-node ~]# /usr/sbin/ntpdate ntp1.aliyun.com

22 Sep 22:57:04 ntpdate[14520]: adjust time server 120.25.115.20 offset -0.001107 sec

[root@master-node ~]# echo "/usr/sbin/ntpdate ntp1.aliyun.com" >>/etc/rc.local

[root@master-node ~]# chmod +x /etc/rc.local

* **所有节点安装JDK1.8（在此只展示master-node节点的配置）；**

[root@master-node ~]# yum -y install java-1.8.0-openjdk

[root@master-node ~]# java -version

openjdk version "1.8.0\_131"

OpenJDK Runtime Environment (build 1.8.0\_131-b12)

OpenJDK 64-Bit Server VM (build 25.131-b12, mixed mode)

* **所有节点安装elasticsearch（在此只展示master-node节点的配置）；**

[root@master-node ~]# rpm --import https://artifacts.elastic.co/GPG-KEY-elasticsearch

[root@master-node ~]# cat <<END >>/etc/yum.repos.d/elastic.repo

[elasticsearch-6.x]

name=Elasticsearch repository for 6.x packages

baseurl=https://artifacts.elastic.co/packages/6.x/yum

gpgcheck=1

gpgkey=https://artifacts.elastic.co/GPG-KEY-elasticsearch

enabled=1

autorefresh=1

type=rpm-md

END

[root@master-node ~]# yum install -y elasticsearch

[root@master-node ~]# rpm -qc elasticsearch

/etc/elasticsearch/elasticsearch.yml ##主配置文件

/etc/elasticsearch/jvm.options

/etc/elasticsearch/log4j2.properties

/etc/elasticsearch/role\_mapping.yml

/etc/elasticsearch/roles.yml

/etc/elasticsearch/users

/etc/elasticsearch/users\_roles

/etc/init.d/elasticsearch

/etc/sysconfig/elasticsearch

/usr/lib/sysctl.d/elasticsearch.conf

/usr/lib/systemd/system/elasticsearch.service

[root@master-node ~]# ll /etc/elasticsearch/

总用量 28

-rw-rw---- 1 root elasticsearch 207 9月 22 02:31 elasticsearch.keystore

-rw-rw---- 1 root elasticsearch 3089 9月 22 06:09 elasticsearch.yml

-rw-rw---- 1 root elasticsearch 3009 9月 14 06:22 jvm.options

-rw-rw---- 1 root elasticsearch 6380 9月 14 06:22 log4j2.properties

-rw-rw---- 1 root elasticsearch 473 9月 14 06:22 role\_mapping.yml

-rw-rw---- 1 root elasticsearch 197 9月 14 06:22 roles.yml

-rw-rw---- 1 root elasticsearch 0 9月 14 06:22 users

-rw-rw---- 1 root elasticsearch 0 9月 14 06:22 users\_roles

* **配置es的master主节点；**

[root@master-node ~]# vi /etc/elasticsearch/elasticsearch.yml ##追加，注意配置项:后的空格，

cluster.name: linuxfan ##集群名称

node.name: master-node ##该节点名称

node.master: true ##该节点为主节点

node.data: false ##表示这不是数据节点

network.host: 192.168.100.108 ##监听ip

http.port: 9200 ##es服务的端口号

discovery.zen.ping.unicast.hosts: ["192.168.100.108", "192.168.100.107", "192.168.100.106"]

##配置自动发现

:wq

* **配置es的data1节点：**

[root@data-node1 ~]# vi /etc/elasticsearch/elasticsearch.yml

cluster.name: linuxfan

node.name: data-node1

node.master: false

node.data: true

network.host: 192.168.100.107

http.port: 9200

discovery.zen.ping.unicast.hosts: ["192.168.100.108", "192.168.100.107", "192.168.100.106"]

:wq

* **配置es的data2节点：**

[root@data-node2 ~]# vi /etc/elasticsearch/elasticsearch.yml

cluster.name: linuxfan

node.name: data-node2

node.master: false

node.data: true

network.host: 192.168.100.106

http.port: 9200

discovery.zen.ping.unicast.hosts: ["192.168.100.108", "192.168.100.107", "192.168.100.106"]

:wq

* **配置es主节点开启es服务：**

[root@master-node ~]# vi /etc/sysctl.conf

vm.max\_map\_count = 655360

[root@master-node ~]# sysctl -p

[root@master-node ~]# systemctl start elasticsearch

[root@master-node ~]# systemctl enable elasticsearch

Created symlink from /etc/systemd/system/multi-user.target.wants/elasticsearch.service to /usr/lib/systemd/system/elasticsearch.service.

注解：如若无法启动：

vi /etc/security/limits.conf

\* soft nofile 65536

\* hard nofile 65536

\* soft nproc 65536

\* hard nproc 65536

[root@master-node ~]# netstat -utpln |grep java

tcp 0 0 192.168.100.108:9200 0.0.0.0:\* LISTEN 14027/java

tcp 0 0 192.168.100.108:9300 0.0.0.0:\* LISTEN 14027/java

注解：9300端口是es集群通信用的，9200则是数据传输时用的；

* **配置es从节点启动es服务：**

[root@data-node1 ~]# systemctl start elasticsearch

[root@data-node1 ~]# systemctl enable elasticsearch

Created symlink from /etc/systemd/system/multi-user.target.wants/elasticsearch.service to /usr/lib/systemd/system/elasticsearch.service.

[root@data-node1 ~]# netstat -utpln|grep java

tcp 0 0 192.168.100.107:9200 0.0.0.0:\* LISTEN 13879/java

tcp 0 0 192.168.100.107:9300 0.0.0.0:\* LISTEN 13879/java

[root@data-node2 ~]# systemctl start elasticsearch

[root@data-node2 ~]# systemctl enable elasticsearch

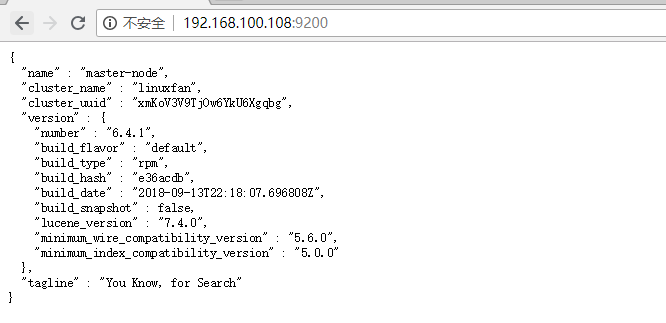
Created symlink from /etc/systemd/system/multi-user.target.wants/elasticsearch.service to /usr/lib/systemd/system/elasticsearch.service.

[root@data-node2 ~]# netstat -utpln|grep java

tcp 0 0 192.168.100.106:9200 0.0.0.0:\* LISTEN 13879/java

tcp 0 0 192.168.100.106:9300 0.0.0.0:\* LISTEN 13879/java

* **在es的master主节点验证集群状态：**



[root@master-node ~]# curl '192.168.100.108:9200/\_cluster/health?pretty'

{

"cluster\_name" : "linuxfan",

"status" : "green",

"timed\_out" : false,

"number\_of\_nodes" : 3,

"number\_of\_data\_nodes" : 2,

"active\_primary\_shards" : 0,

"active\_shards" : 0,

"relocating\_shards" : 0,

"initializing\_shards" : 0,

"unassigned\_shards" : 0,

"delayed\_unassigned\_shards" : 0,

"number\_of\_pending\_tasks" : 0,

"number\_of\_in\_flight\_fetch" : 0,

"task\_max\_waiting\_in\_queue\_millis" : 0,

"active\_shards\_percent\_as\_number" : 100.0

}

注解：curl操作es集群更多操作：<http://zhaoyanblog.com/archives/732.html>

* **在es集群的master主节点安装并配置启动kibana：**

[root@master-node ~]# yum -y install kibana

[root@master-node ~]# vi /etc/kibana/kibana.yml ##追加

server.port: 5601

server.host: 192.168.100.108

elasticsearch.url: "http://192.168.100.108:9200"

logging.dest: /var/log/kibana.log

:wq

[root@master-node ~]# touch /var/log/kibana.log

[root@master-node ~]# chmod 777 /var/log/kibana.log

[root@master-node ~]# systemctl start kibana

[root@master-node ~]# systemctl enable kibana

Created symlink from /etc/systemd/system/multi-user.target.wants/kibana.service to /etc/systemd/system/kibana.service.

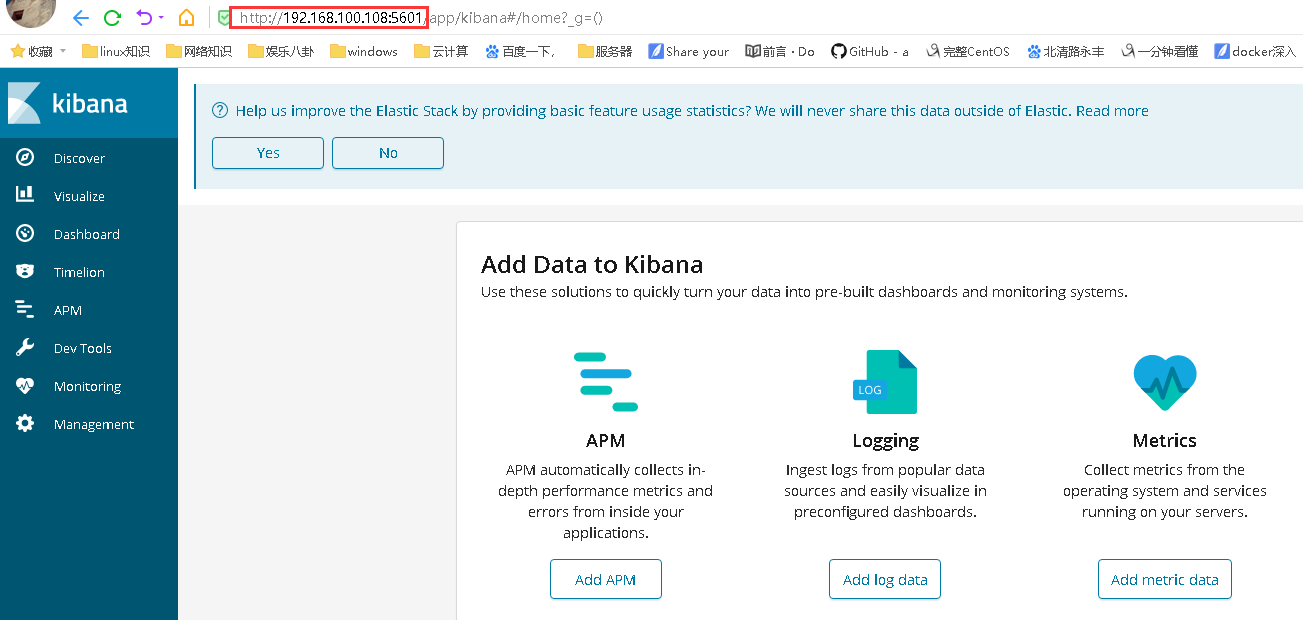
[root@master-node ~]# netstat -utpln |grep node

tcp 0 0 192.168.100.108:5601 0.0.0.0:\* LISTEN 14294/node

注：由于kibana是使用node.js开发的，所以进程名称为node

* **使用web浏览器访问kibana的web页面进行验证：**

在浏览器里进行访问http://192.168.100.108:5601 ，由于我们并没有安装x-pack，所以此时是没有用户名和密码的，可以直接访问的：



* **在es集群的data1从节点安装并测试配置启动logstash；**

[root@data-node1 ~]# yum -y install logstash

[root@data-node1 ~]# vi /etc/logstash/conf.d/syslog.conf ##追加

input { ##定义日志源

syslog {

type => "system-syslog" ##定义类型

port => 10514 ##定义监听端口

}

}

output { ##定义日志输出

stdout {

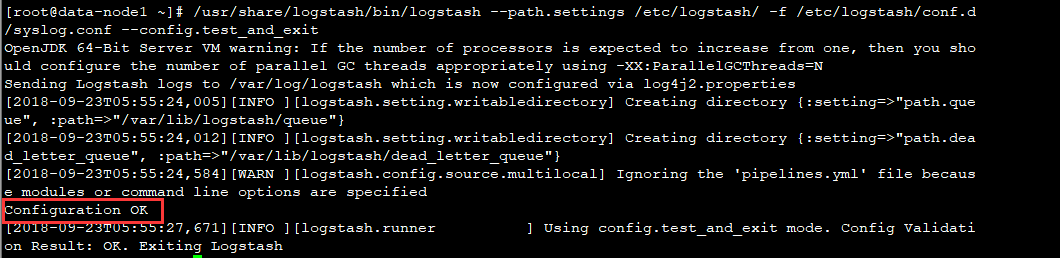
codec => rubydebug ##将日志输出到当前的终端上显示

}

}

:wq

[root@data-node1 ~]# /usr/share/logstash/bin/logstash --path.settings /etc/logstash/ -f /etc/logstash/conf.d/syslog.conf --config.test\_and\_exit ##检查配置文件，注意内存的使用，如若内存不足，会出现检查jdk失败；



注解：

--path.settings ##用于指定logstash的配置文件所在的目录

-f ##指定需要被检测的配置文件的路径

--config.test\_and\_exit ##指定检测完之后就退出，不然就会直接启动了

[root@data-node1 ~]# vi /etc/rsyslog.conf ##追加

\*.\* @@192.168.100.107:10514

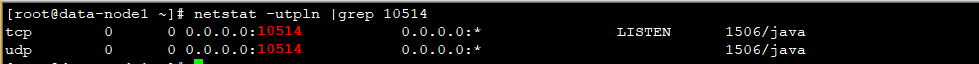
:wq

[root@data-node1 ~]# systemctl restart rsyslog

[root@data-node1 ~]# /usr/share/logstash/bin/logstash --path.settings /etc/logstash/ -f /etc/logstash/conf.d/syslog.conf ##启动logstash

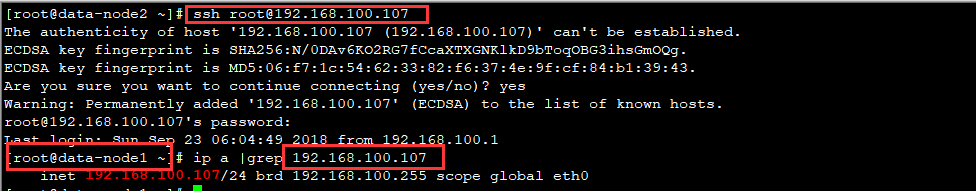
这时终端会停留在这里，因为我们在配置文件中定义的是将信息输出到当前终端；

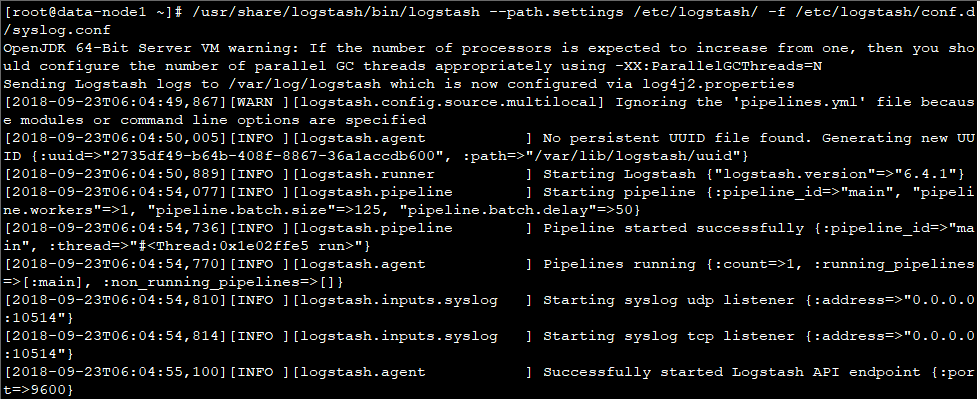
打开新终端检查一下10514端口是否已被监听；

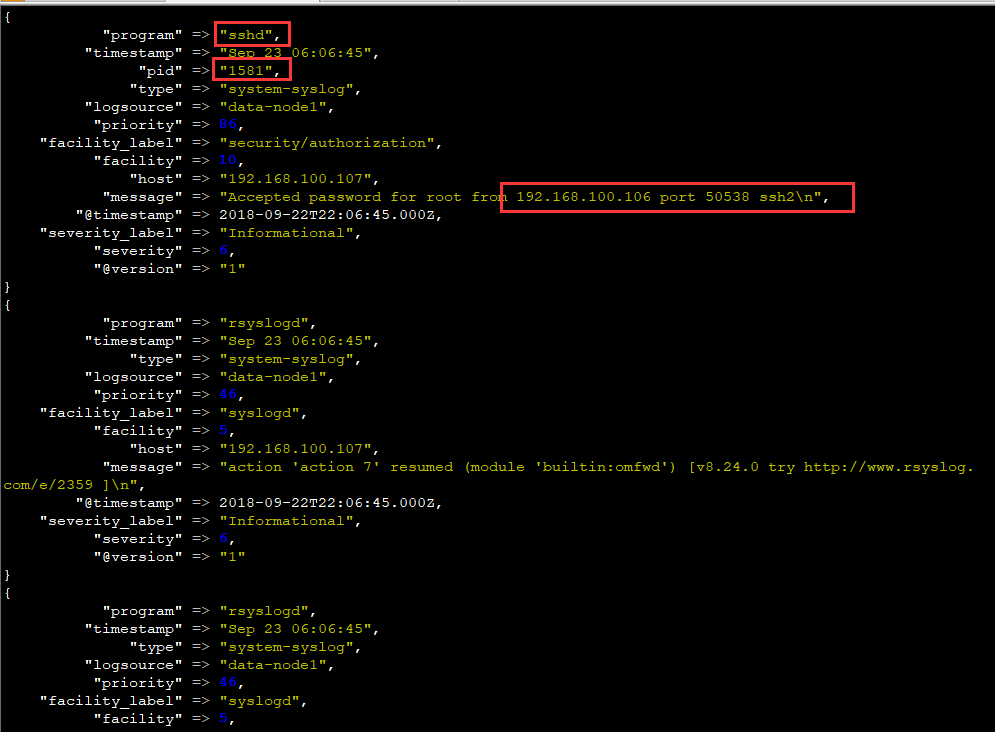




然后在别的机器ssh登录到这台机器上，测试一下有没有日志输出；







如上，可以看到，终端中以JSON的格式打印了收集到的日志，测试成功。

* **配置logstash服务让收集的日志信息输出到es服务器中，而不是当前终端：**

[root@data-node1 ~]# vi /etc/logstash/conf.d/syslog.conf ##更改为如下内容

input {

syslog {

type => "system-syslog"

port => 10514

}

}

output {

elasticsearch {

hosts => ["192.168.100.108:9200"]

index => "system-syslog-%{+YYYY.MM}"

}

}

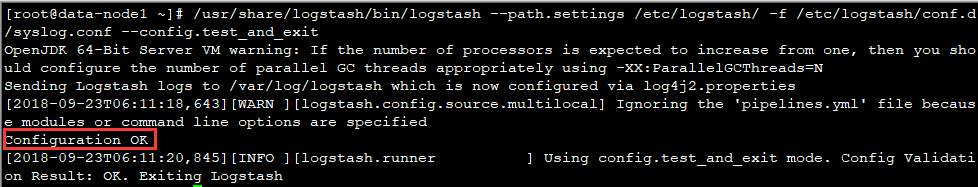
:wq

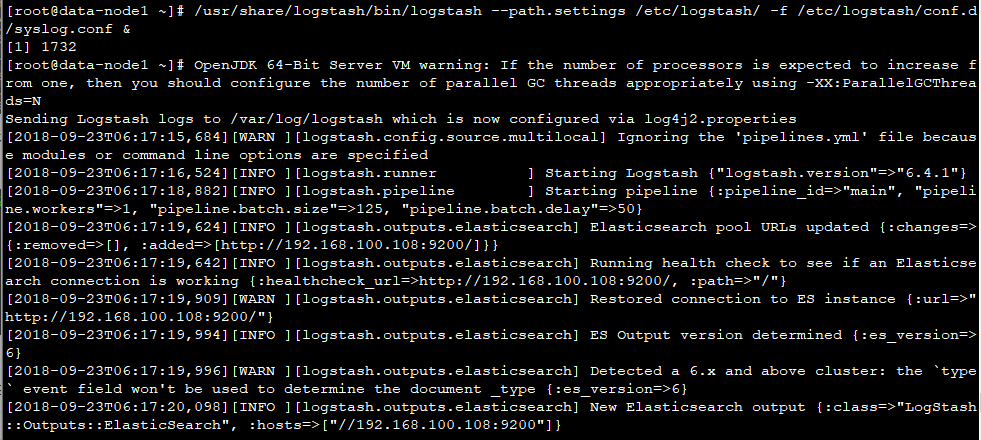
[root@data-node1 ~]# vi /etc/logstash/logstash.yml

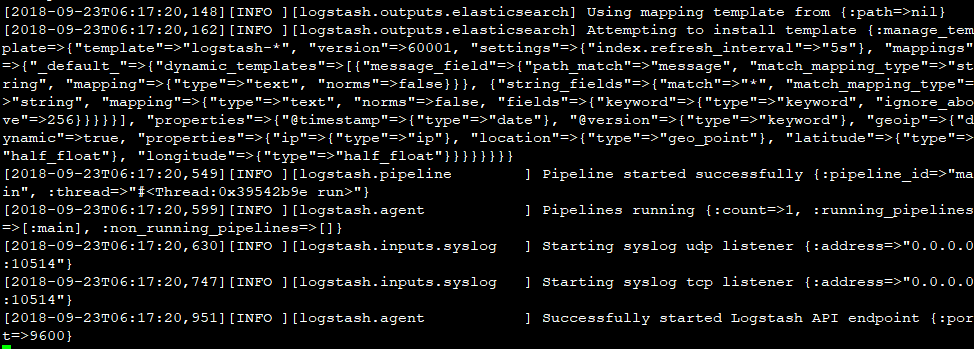
http.host: "192.168.100.107"

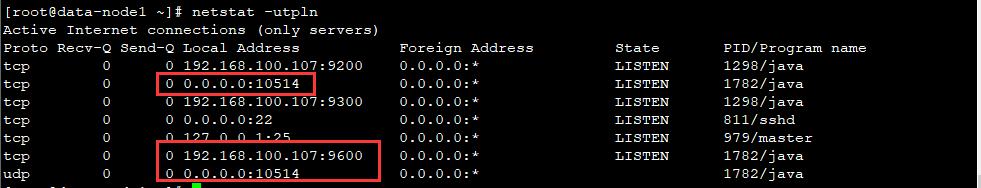
:wq

[root@data-node1 ~]# /usr/share/logstash/bin/logstash --path.settings /etc/logstash/ -f /etc/logstash/conf.d/syslog.conf --config.test\_and\_exit

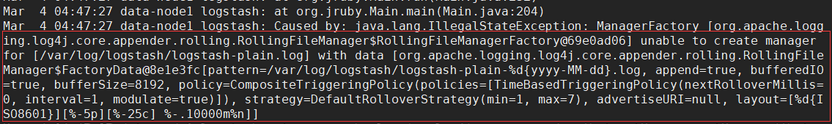


[root@data-node1 ~]# /usr/share/logstash/bin/logstash --path.settings /etc/logstash/ -f /etc/logstash/conf.d/syslog.conf & 



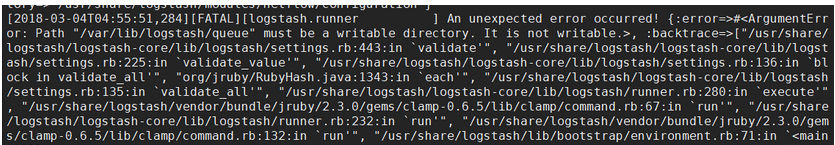


注解：启动logstash后，进程是正常存在的，但是9600以及10514端口却没有被监听。检查message日志文件发现：

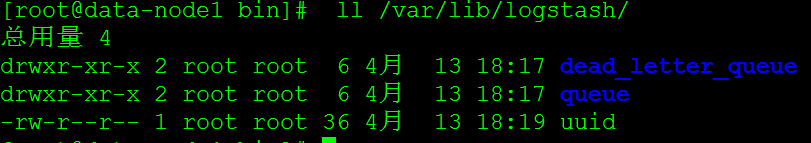


[root@data-node1 ~]# chown logstash /var/log/logstash/logstash-plain.log

再次检查日志文件：

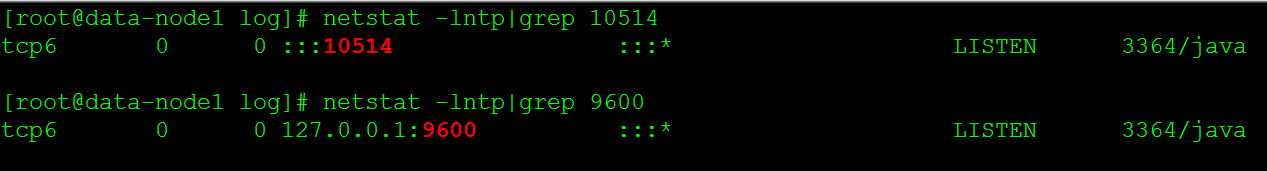


可以看到，依旧是权限的问题，这是因为之前我们以root的身份在终端启动过logstash，所以产生的相关文件的属组属主都是root，同样的，也是设置一下权限即可：



[root@data-node1 bin]# chown -R logstash /var/lib/logstash/

这次就没问题了，端口正常监听了，这样我们的logstash服务就启动成功了：

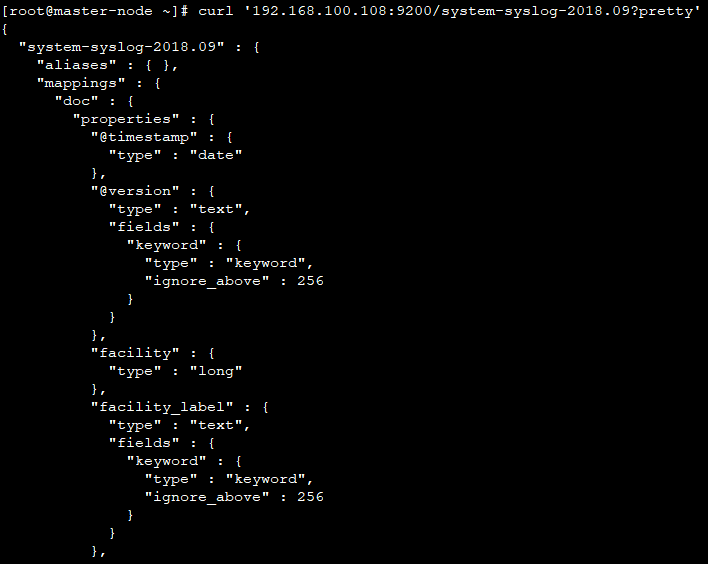


* **访问kibana服务器查看日志信息：**

[root@master-node ~]# curl '192.168.100.108:9200/\_cat/indices?v' ##查看索引信息，索引信息正常说明logstash服务于es服务通信正常，如若无法正常生成索引，需要新建数据连接到logstash服务器；

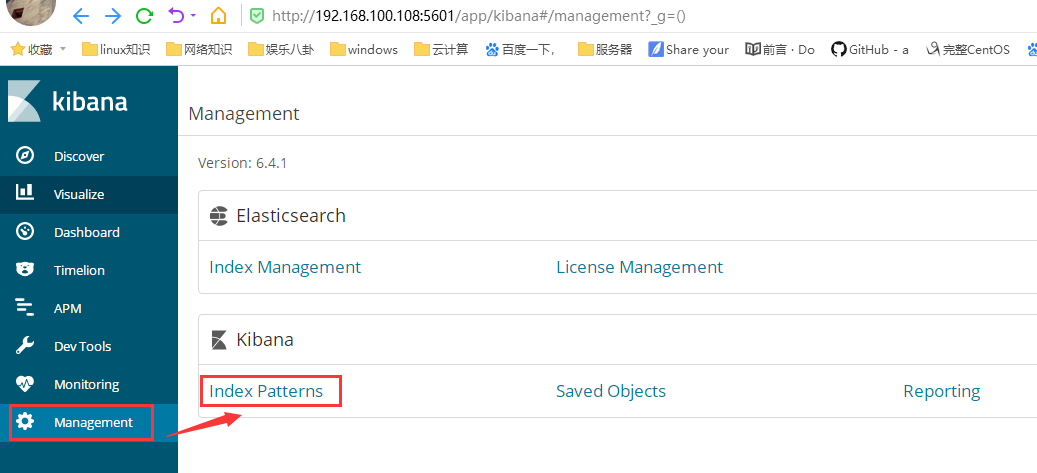


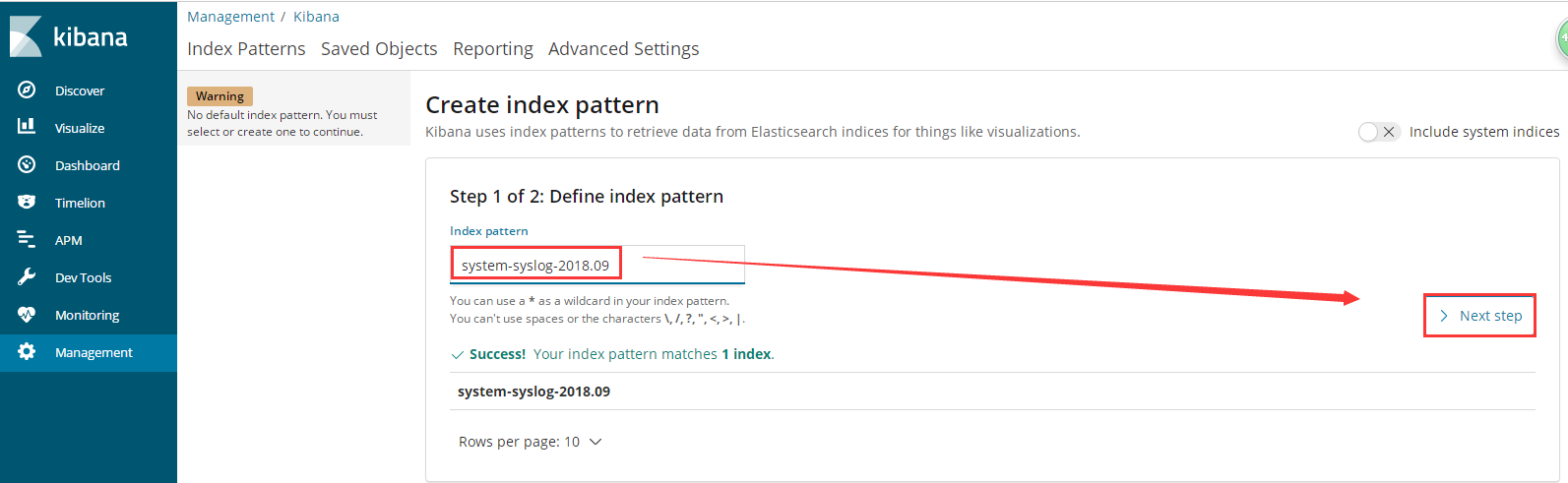
[root@master-node ~]# curl '192.168.100.108:9200/system-syslog-2018.09?pretty' ##查看索引的详细信息



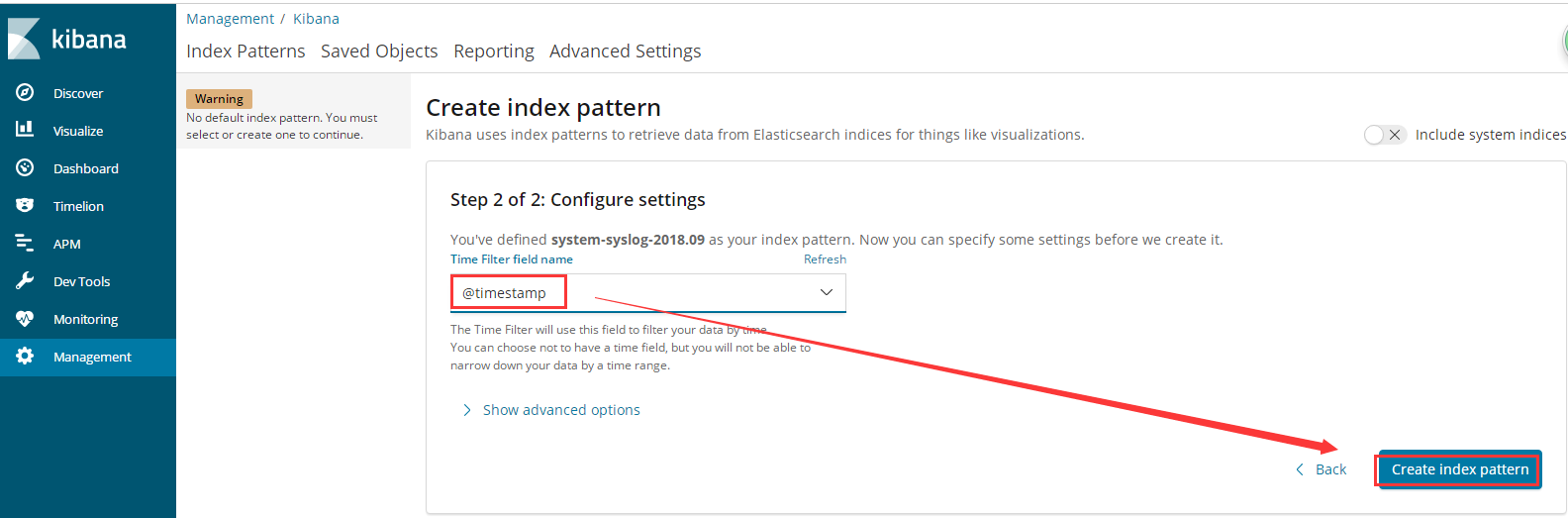
[root@master-node ~]# curl -XDELETE '192.168.200.1:9200/system-syslog-2018.04' ##删除索引

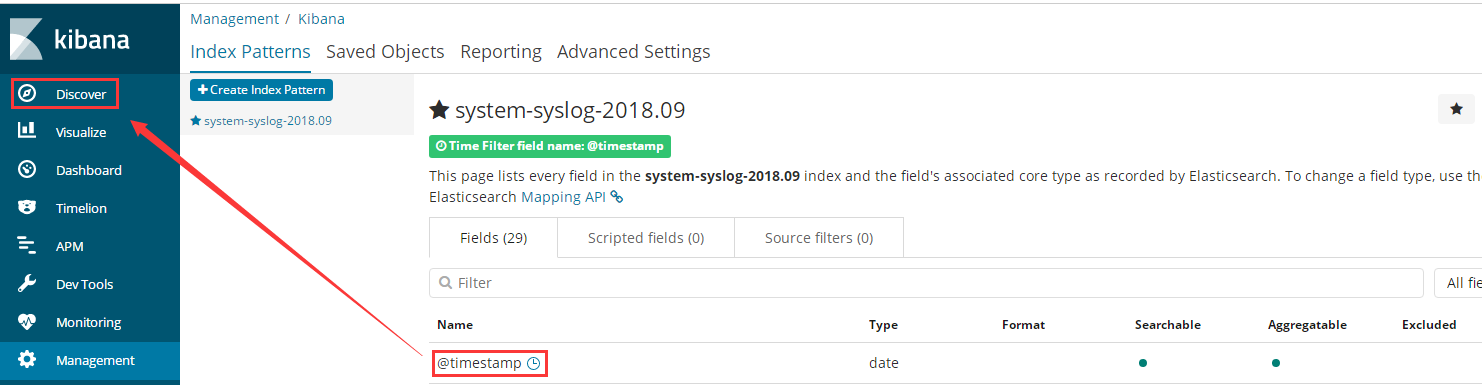
* **配置kibana的web网页设置索引：**



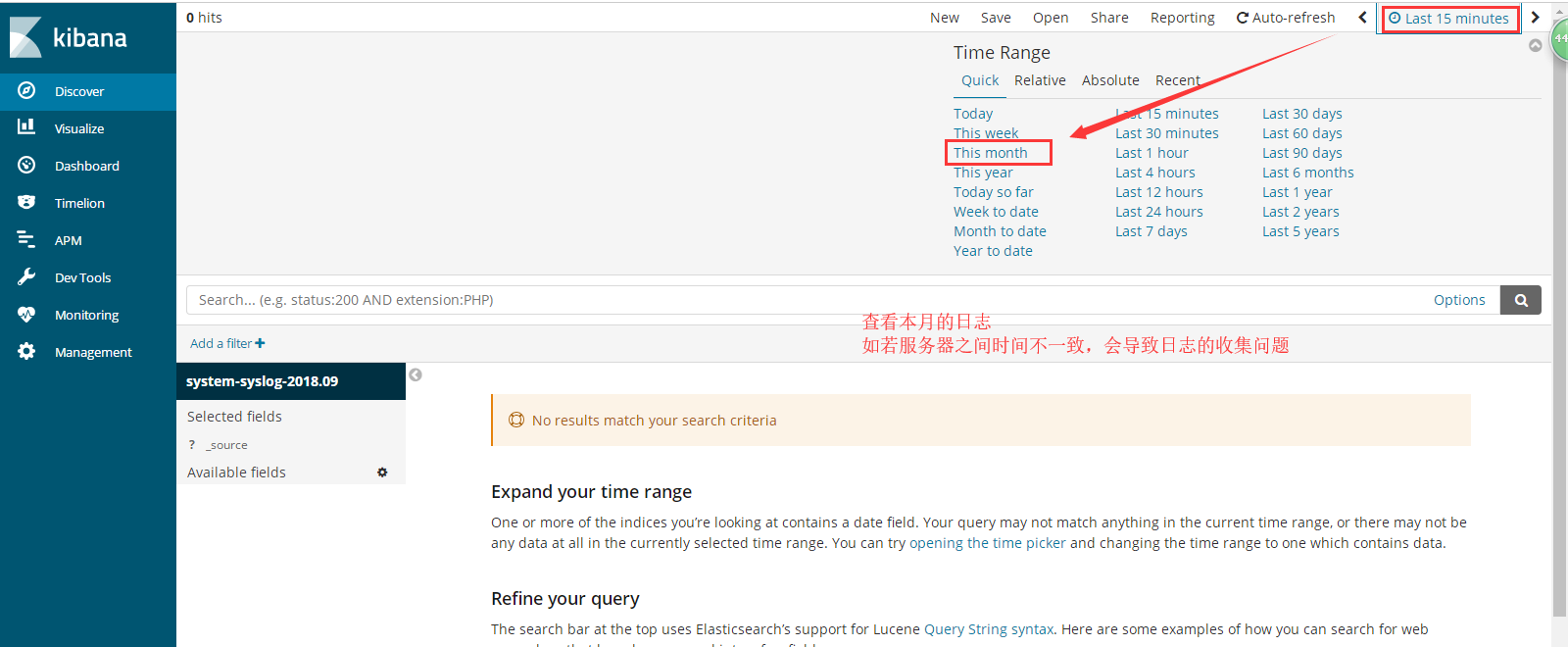


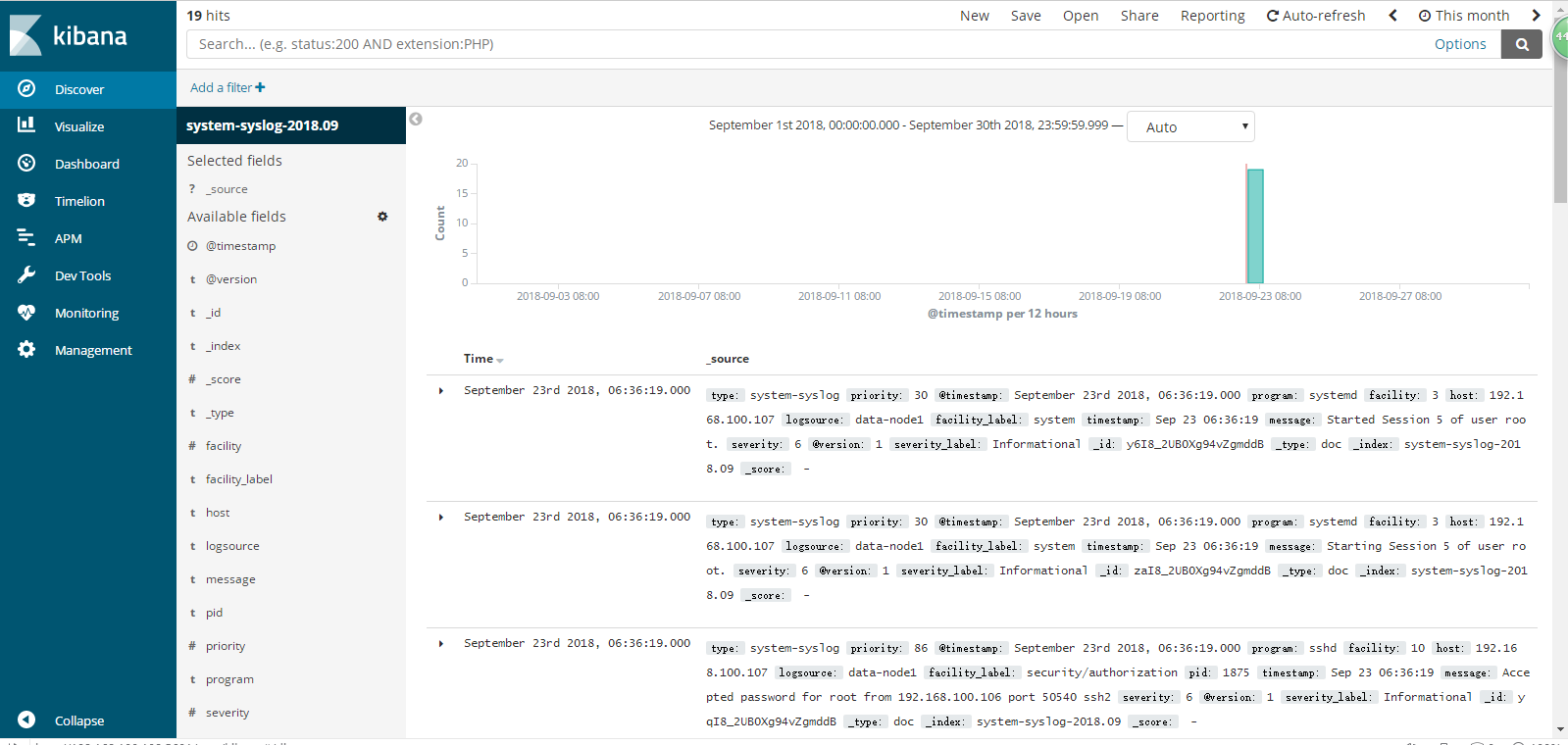


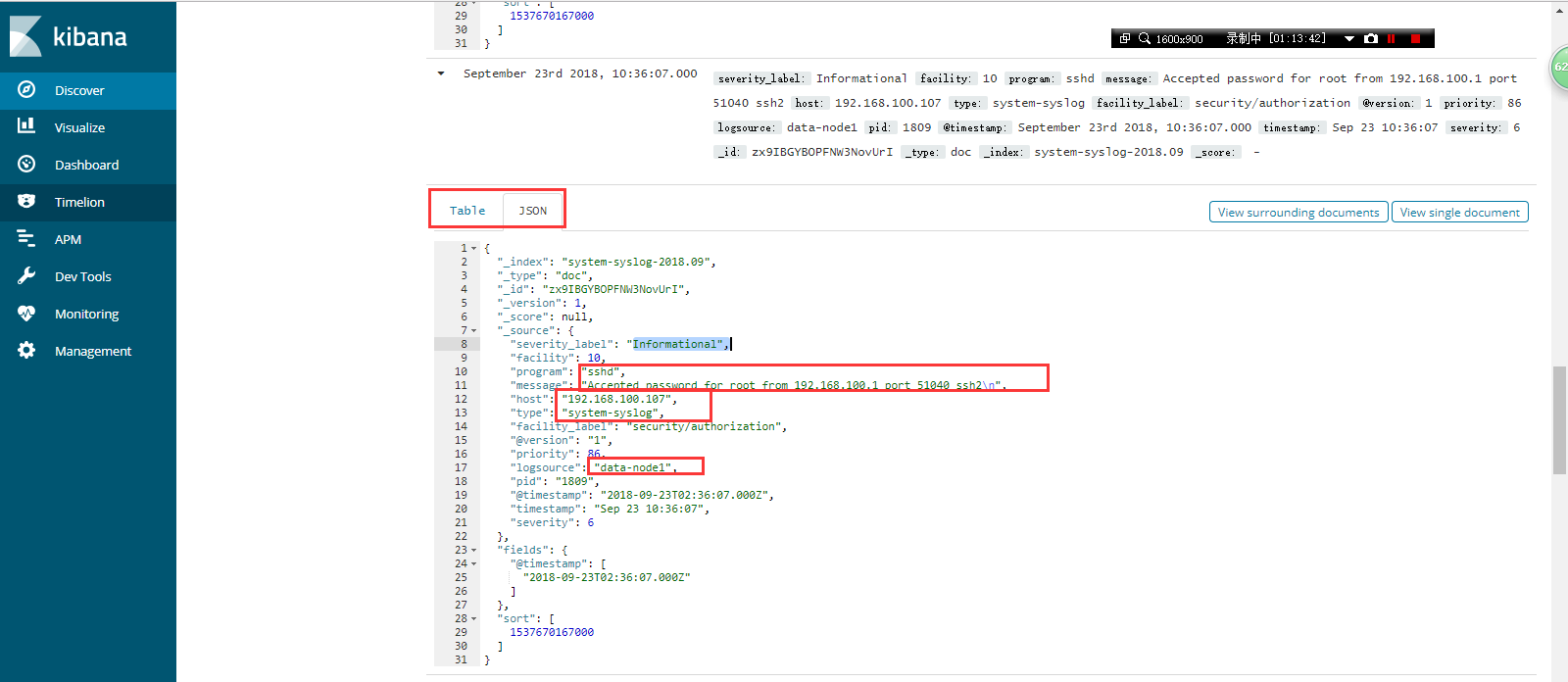




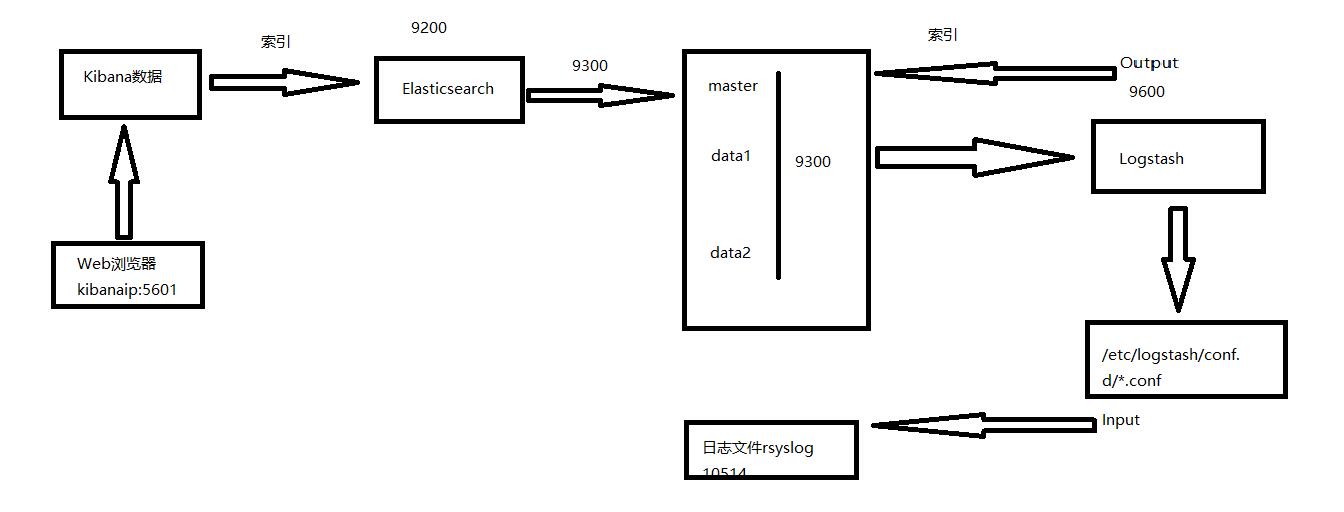
* **配置kibana的web页面显示日志收集信息：**







* **总结流程：**



**五、案例：应用ELK日志分析平台进行分析nginx日志：**

[root@nginx ~]# wget http://nginx.org/packages/centos/7/x86\_64/RPMS/nginx-1.12.2-1.el7\_4.ngx.x86\_64.rpm

[root@nginx ~]# rpm -ivh nginx-1.12.2-1.el7\_4.ngx.x86\_64.rpm --nodeps

[root@nginx ~]# systemctl start nginx

[root@nginx ~]# systemctl enable nginx

Created symlink from /etc/systemd/system/multi-user.target.wants/nginx.service to /usr/lib/systemd/system/nginx.service.

[root@nginx ~]# netstat -utpln |grep 80

tcp 0 0 0.0.0.0:80 0.0.0.0:\* LISTEN 1163/nginx: master



[root@nginx ~]# cat <<END >>/etc/yum.repos.d/elastic.repo

[elasticsearch-6.x]

name=Elasticsearch repository for 6.x packages

baseurl=https://artifacts.elastic.co/packages/6.x/yum

gpgcheck=1

gpgkey=https://artifacts.elastic.co/GPG-KEY-elasticsearch

enabled=1

autorefresh=1

type=rpm-md

END

[root@nginx ~]# yum -y install filebeat

[root@nginx ~]# rpm -qc filebeat

/etc/filebeat/filebeat.yml

/etc/filebeat/modules.d/apache2.yml.disabled

/etc/filebeat/modules.d/auditd.yml.disabled

/etc/filebeat/modules.d/elasticsearch.yml.disabled

/etc/filebeat/modules.d/icinga.yml.disabled

/etc/filebeat/modules.d/iis.yml.disabled

/etc/filebeat/modules.d/kafka.yml.disabled

/etc/filebeat/modules.d/kibana.yml.disabled

/etc/filebeat/modules.d/logstash.yml.disabled

/etc/filebeat/modules.d/mongodb.yml.disabled

/etc/filebeat/modules.d/mysql.yml.disabled

/etc/filebeat/modules.d/nginx.yml.disabled

/etc/filebeat/modules.d/osquery.yml.disabled

/etc/filebeat/modules.d/postgresql.yml.disabled

/etc/filebeat/modules.d/redis.yml.disabled

/etc/filebeat/modules.d/system.yml.disabled

/etc/filebeat/modules.d/traefik.yml.disabled

27 paths:

28 - /var/log/\*.log

29 - /var/log/nginx/access.log

