**ELK集群部署及收集日志**

**一、ELK说明**

ELK Stack 是 Elasticsearch、Logstash、Kibana 三个开源软件的组合。在实时数据检索和分析场合，三者通常是配合共用，而且又都先后归于 Elastic.co 公司名下，故有此简称。

ELK Stack 在最近两年迅速崛起，成为机器数据分析，或者说实时日志处理领域，开源界的第一选择。和传统的日志处理方案相比，ELK Stack 具有如下几个优点：

处理方式灵活。Elasticsearch 是实时全文索引，不需要像 storm 那样预先编程才能使用；

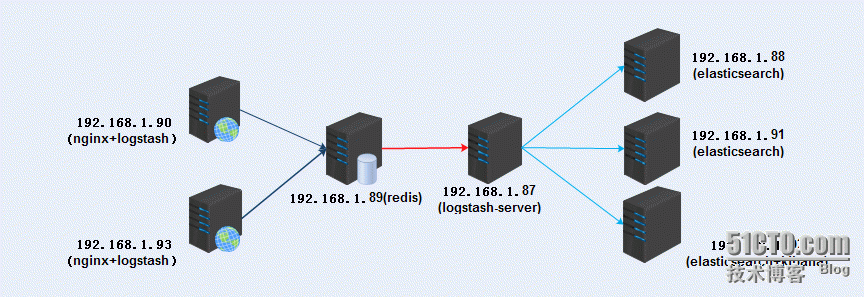
配置简易上手。Elasticsearch 全部采用 JSON 接口，Logstash 是 Ruby DSL 设计，都是目前业界最通用的配置语法设计；

检索性能高效。虽然每次查询都是实时计算，但是优秀的设计和实现基本可以达到全天数据查询的秒级响应；

集群线性扩展。不管是 Elasticsearch 集群还是 Logstash 集群都是可以线性扩展的；

前端操作炫丽。Kibana 界面上，只需要点击鼠标，就可以完成搜索、聚合功能，生成炫丽的仪表板。

**二、架构图**



**三、规划说明**

logstash收集数据:

192.168.113.121(nginx+logstash)

192.168.113.118(nginx+logstash)

redis提供队列:

192.168.113.121(redis)

logstash-server:

192.168.113.120(logstash[server])

elasticsearch集群:

192.168.113.120(elasticsearch)

192.168.113.121 (elasticsearch+ kibana)所有服务器时间同步：

[root@t2 ~]# crontab -e

\*/1 \* \* \* \* /usr/sbin/ntpdate 202.108.6.95 &>/dev/null

软件版本：

elasticsearch-1.7.2

logstash-6.1.1

jdk1.8.0\_152

kibana-4.1.2-linux-x64

redis

**四、安装部署nginx+logstash**

192.168.113.120,192.168.113.121

1.Nginx安装步骤略

2.安装jdk

tar -xf jdk-8u91-linux-x64.tar.gz -C /usr/local/

[root@t2 local]# vim /etc/profile.d/java.sh

JAVA\_HOME=/usr/local/jdk1.8.0\_152

JRE\_HOME=$JAVA\_HOME/jre

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

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

export JAVA\_HOME JRE\_HOME PATH CLASSPATH

chmod +x /etc/profile.d/java.sh

[root@t2 local]# sh /etc/profile.d/java.sh

3. 安装logstsh

[root@t1 ~]# tar xf logstash-6.1.1.tar.gz -C /usr/local/

[root@t2 config]# vim /usr/local/logstash-6.1.1/config/nginx\_to\_redis.log.conf

input {

file {

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

type => "ttt"

start\_position => "beginning"

}

}

filter{

grok {

match => { "message" => "%{COMBINEDAPACHELOG}" }

remove\_field => ["message"]

}

}

output {

redis {

host => "192.168.113.121"

key => 'logstash-redis'

data\_type => 'list'

}

}

启动: nohup /usr/local/logstash-6.1.1/bin/logstash -f /usr/local/logstash-6.1.1/config/ nginx\_to\_redis.log.conf &

**五、安装部署redis**

192.168.113.121(redis)

5.1.下载redis官网下载

tar -xf redis-3.0.7.tar.gz

cd redis-3.0.7

make

mkdir -p /usr/local/redis/{conf,bin}

cp \*.conf /usr/local/redis/conf/

cp runtest\* /usr/local/redis/

cp mkreleasehdr.sh redis-benchmark redis-check-aof redis-check-dump redis-cli redis-sentinel redis-server redis-trib.rb /usr/local/redis/bin/

mkdir -pv /data/redis/db

mkdir /data/log/redis –pv

安装epel 源

yum -y install redis

5.2.启动redis

Service redis start

**六、安装部署logstash server**

192.168.113.120(logstash[server])

[root@t1 ~]# tar xf logstash-6.1.1.tar.gz -C /usr/local/

root@t2 config]# vim /usr/local/logstash-6.1.1/config/redis\_to\_elk.conf

input {

redis {

port => "6379"

host => "192.168.113.121"

data\_type => "list"

key => "logstash-redis"

type => "redis-input"

}

}

output {

elasticsearch {

hosts => "192.168.113.120"

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

}

}

启动: nohup /usr/local/logstash-6.1.1/bin/logstash -f /usr/local/logstash-6.1.1/config/ nginx\_to\_redis.log.conf &

**七、安装部署elasticsearch集群**

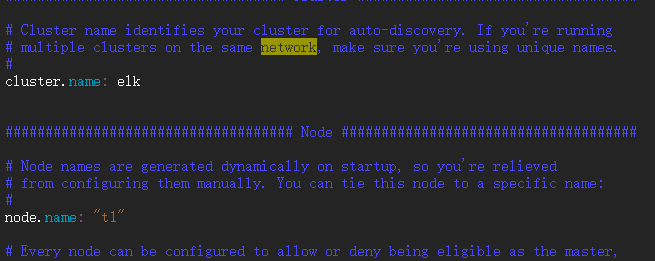
**192.168.113.120 192.168.113.121**

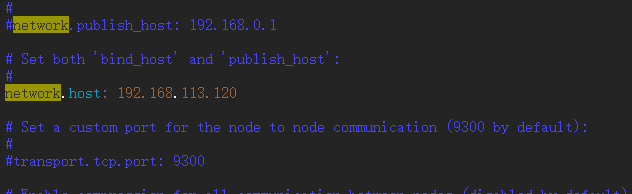
**安装jdk同上**

**安装elasticsearch**

**[root@t1 elk]# tar xf elasticsearch-1.7.2.tar.gz -C /usr/local/**

[root@t1 elk]# vim /usr/local/elasticsearch-1.7.2/config/elasticsearch.yml





启动：

[root@t1 bin]# vim /usr/local/elasticsearch-1.7.2/bin/elk\_start.sh

#!/bin/bash

#

/usr/bin/nohup /usr/local/elasticsearch-1.7.2/bin/elasticsearch > /var/log/logstash.log 2>&1 &

[root@t1 bin]# chmod +x elk\_start.sh

[root@t1 bin]# sh /usr/local/elasticsearch-1.7.2/bin/elk\_start.sh

启动两个端口

9200：集群之间事务通信

9300：集群选举等

1.查看健康信息：

# curl -XGET 'http://192.168.1.88:9200/\_cluster/health?pretty'

{

"cluster\_name" : "qm.ela.com",

"status" : "green",

"timed\_out" : false,

"number\_of\_nodes" : 3,

"number\_of\_data\_nodes" : 3,

"active\_primary\_shards" : 14,

"active\_shards" : 28,

"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

}

2.查看节点数：

# curl -XGET '192.168.1.88:9200/\_cat/nodes?v'

host ip heap.percent ram.percent load node.role master name

192.168.1.91 192.168.1.91 10 87 0.00 d m n2.qm.com

192.168.1.92 192.168.1.92 7 93 0.00 d m n3.qm.com

192.168.1.88 192.168.1.88 17 88 0.00 d \* n1.qm.com

由此可见，192.168.1.88已经成为主节点

3.查看节点分片信息：

# curl -XGET '192.168.1.88:9200/\_cat/indices?v'

health status index pri rep docs.count docs.deleted store.size pri.store.size

green open .kibana 1 1 3 0 49.8kb 24.9kb

green open .marvel-es-data-1 1 1 14 274 131.5kb 63.2kb

green open logstash-2016.05.10 5 1 168 0 543.3kb 271.6kb

green open logstash-2016.05.11 5 1 118 0 315.6kb 161kb

green open .marvel-es-1-2016.05.11 1 1 38433 56 35.6mb 17.8mb

green open .marvel-es-1-2016.05.10 1 1 42622 16 24.5mb

安装插件

1. 安装插件marvel，三个节点安装一致：

[root@t1 bin]# cd /usr/local/elasticsearch-1.7.2/bin

[root@t1 bin]# ./plugin -install elasticsearch/marvel/latest

1. 安装插件Bigdesk 插件，集群节点同步

[root@t1 bin]# ./plugin -install lukas-vlcek/bigdesk/2.5.0

1. 安装head插件，三个节点安装一致：

[root@t1 bin]# ./plugin -install mobz/elasticsearch-head

4.安装kopf插件：

[root@t1 bin]# ./plugin install lmenezes/elasticsearch-kopf

查看安装的插件：

root@t1 bin]# ./plugin list

Installed plugins in /usr/share/elasticsearch/plugins:

    - head

    - bigdesk

    - analysis-kuromoji

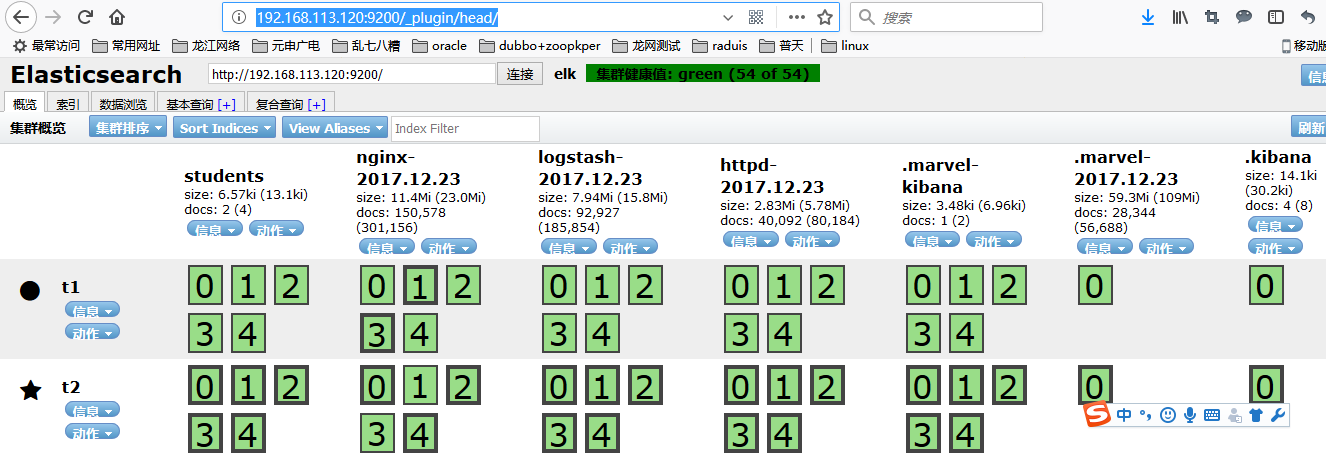
    - kopf

    - license

    - marvel-agent

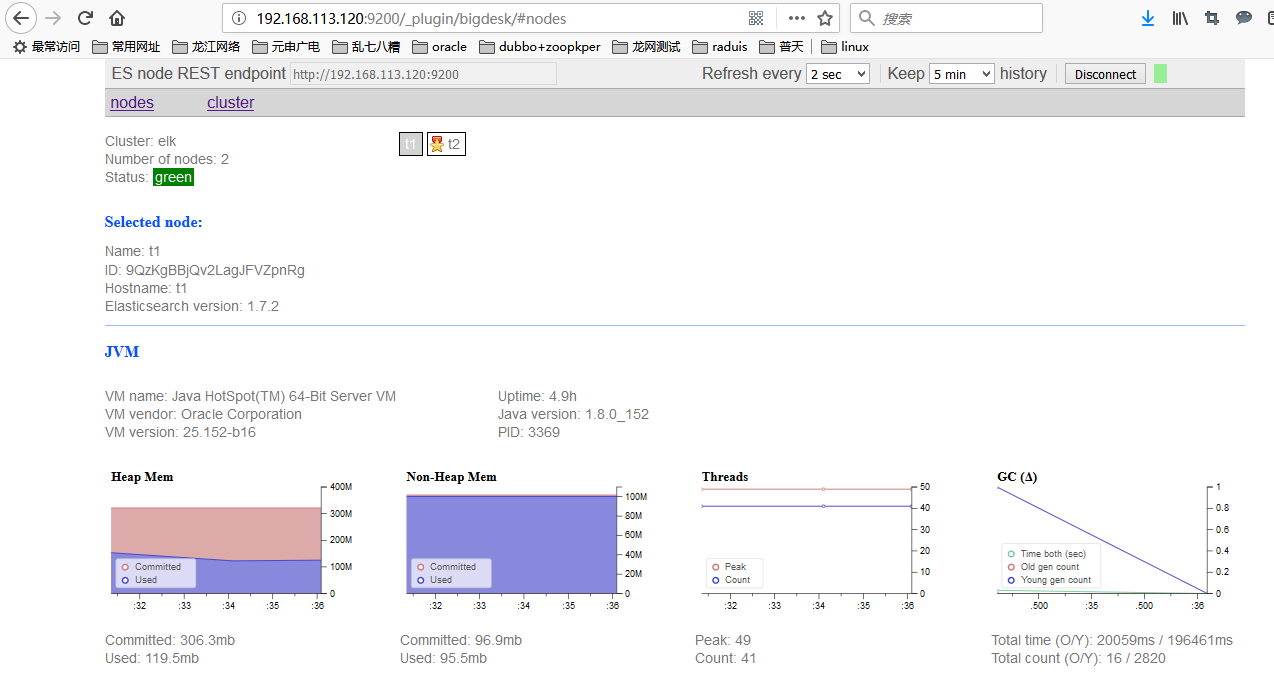
**访问head，查看集群是否一致**

<http://192.168.113.120:9200/_plugin/head/>



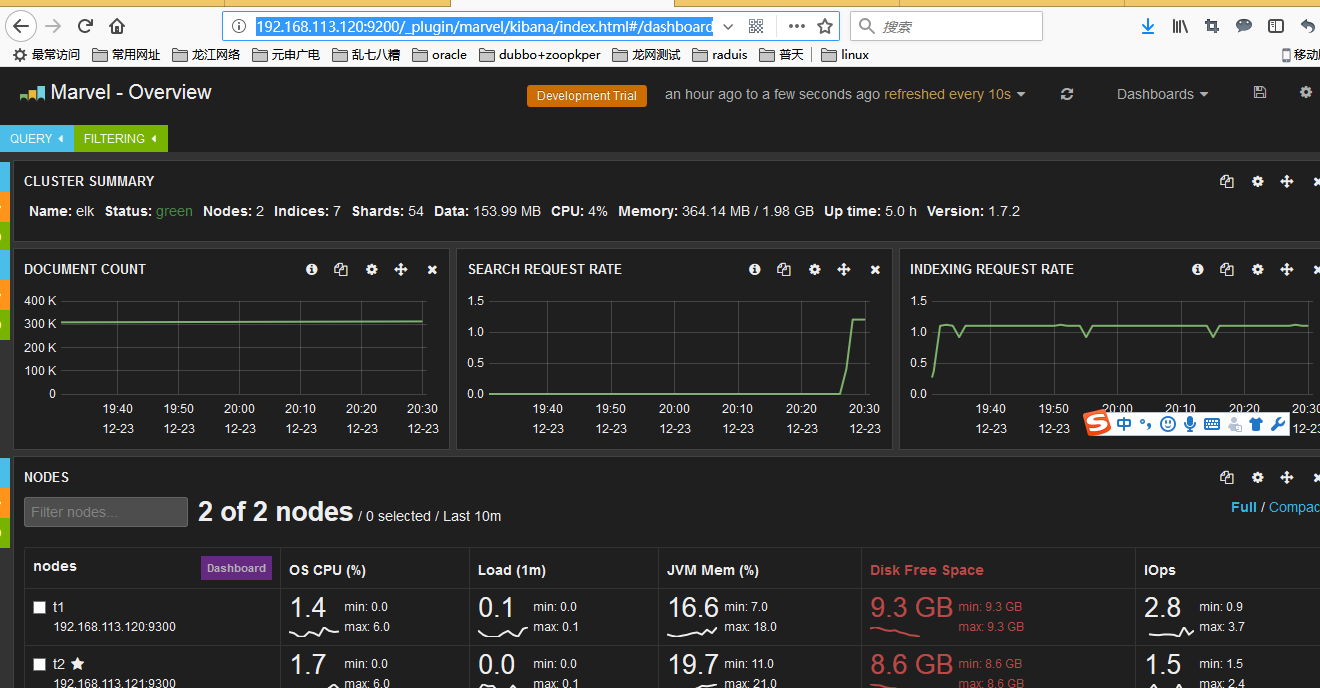
**访问bigdesk，查看信息：**

**http://192.168.113.120:9200/\_plugin/bigdesk/**



**访问marvel，查看信息：**

**http://192.168.113.120:9200/\_plugin/marvel**



**八、安装kibana**

192.168.1.92(kibana)

8.1.下载包：

https://www.elastic.co/downloads/kibana

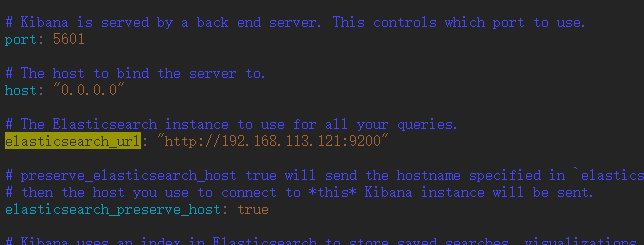
8.2.安装：

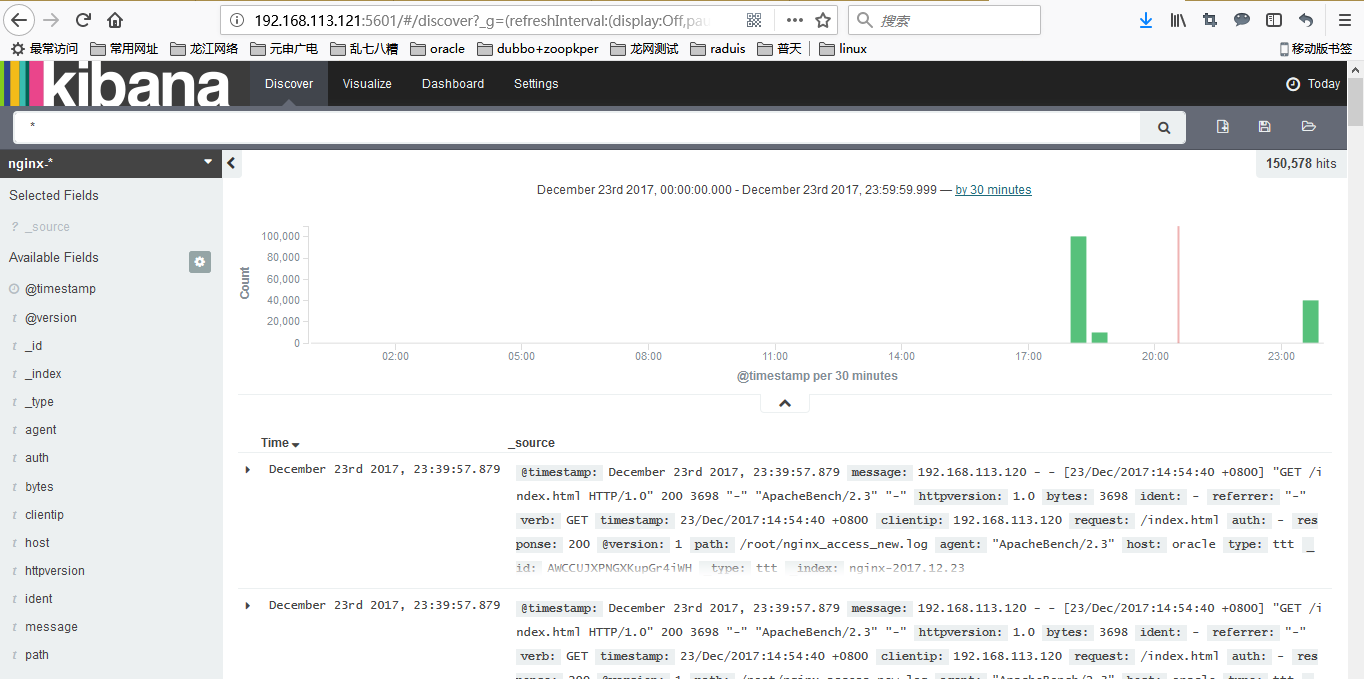
tar -xf kibana-4.5.0-linux-x64.tar.gz -C /usr/local

8.3

配置

[root@t2 config]# vim /usr/local/kibana-4.1.2-linux-x64/config/kibana.ymls



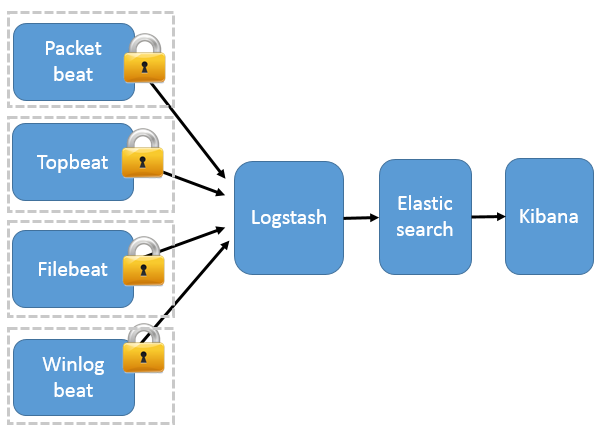


九、Beats 作为日志搜集器

这种架构引入 Beats 作为日志搜集器。目前 Beats 包括四种：

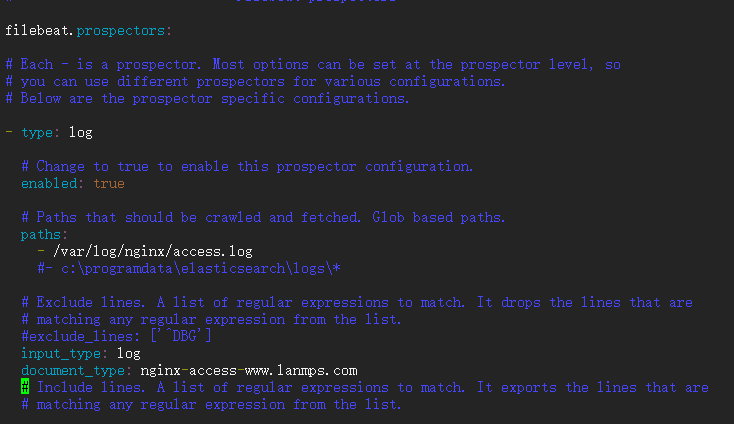
* Packetbeat（搜集网络流量数据）；
* Topbeat（搜集系统、进程和文件系统级别的 CPU 和内存使用情况等数据）；
* Filebeat（搜集文件数据）；
* Winlogbeat（搜集 Windows 事件日志数据）。

Beats 将搜集到的数据发送到 Logstash，经 Logstash 解析、过滤后，将其发送到 Elasticsearch 存储，并由 Kibana 呈现给用户。详见图 3。

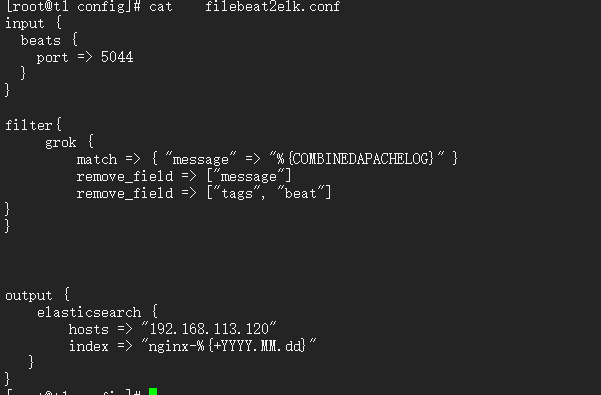


这种架构解决了 Logstash 在各服务器节点上占用系统资源高的问题。相比 Logstash，Beats 所占系统的 CPU 和内存几乎可以忽略不计。另外，Beats 和 Logstash 之间支持 SSL/TLS 加密传输，客户端和服务器双向认证，保证了通信安全。因此这种架构适合对数据安全性要求较高，同时各服务器性能比较敏感的场景

Filebeat 配置文件



Logstash server 配置文件



input {

beats {

port => 5044

}

}

filter{

grok {

match => { "message" => "%{COMBINEDAPACHELOG}" }

remove\_field => ["message"]

remove\_field => ["tags", "beat"]

}

}

output {

elasticsearch {

hosts => "192.168.113.120"

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

}

}