**1、ES安装**

**1.1 下载安装软件**

mkdir -p /data/es\_soft/

cd /data/es\_soft/

wget https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-6.6.0.rpm

rpm -ivh elasticsearch-6.6.0.rpm

**1.2 配置启动**

systemctl daemon-reload

systemctl enable elasticsearch.service

systemctl start elasticsearch.service

systemctl status elasticsearch.service

**1.3 检查是否启动成功**

ps -ef|grep elastic

lsof -i:9200

**1.4 Elasticsearch目录文件说明**

|  |
| --- |
| rpm -ql elasticsearch #查看elasticsearch软件安装了哪些目录  rpm -qc elasticsearch #查看elasticsearch的所有配置文件  /etc/elasticsearch/elasticsearch.yml #配置文件  /etc/elasticsearch/jvm.options #jvm虚拟机配置文件  /etc/init.d/elasticsearch #init启动文件  /etc/sysconfig/elasticsearch #环境变量配置文件  /usr/lib/sysctl.d/elasticsearch.conf #sysctl变量文件，修改最大描述符  /usr/lib/systemd/system/elasticsearch.service #systemd启动文件  /var/lib/elasticsearch # 数据目录  /var/log/elasticsearch #日志目录  /var/run/elasticsearch #pid目录 |

**1.5 修改配置文件(单机)**

|  |
| --- |
| cluster.name: dba5  node.name: node-1  path.data: /data/elasticsearch  path.logs: /var/log/elasticsearch  bootstrap.memory\_lock: true  network.host: 192.168.91.55,127.0.0.1  http.port: 9200 |

**1.6修改配置重新启动**

mkdir /data/elasticsearch

chown -R elasticsearch:elasticsearch /data/elasticsearch/

systemctl restart elasticsearch

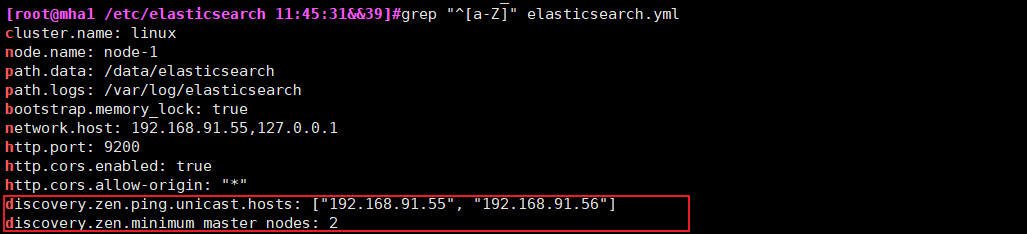
systemctl status elasticsearch

**1.7锁定内存失败解决**

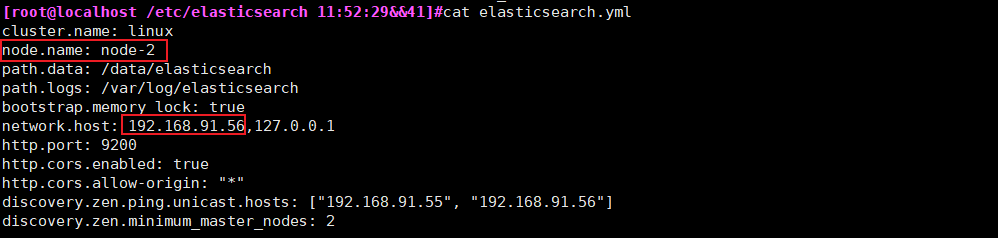
|  |
| --- |
| 官方解决方案  https://www.elastic.co/guide/en/elasticsearch/reference/6.4/setup-configuration-memory.html  https://www.elastic.co/guide/en/elasticsearch/reference/6.4/setting-system-settings.html#sysconfig  ### 修改启动配置文件  vim /usr/lib/systemd/system/elasticsearch.service  ### 增加如下参数  [Service]  LimitMEMLOCK=infinity  ### 重新启动  systemctl daemon-reload  systemctl restart elasticsearch |

**1.8 新增第二个节点**

每个节点都要新增如下配置



新增节点需要修改的地方如下图



Node-1

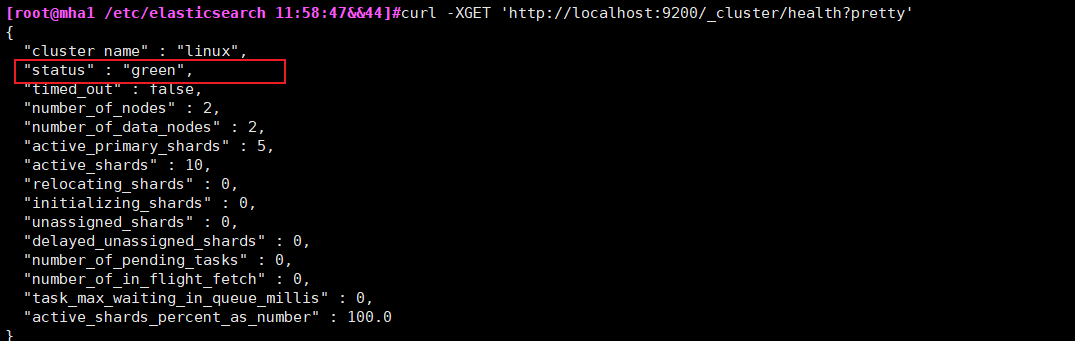
|  |
| --- |
| cluster.name: linux  node.name: node-1  path.data: /data/elasticsearch  path.logs: /var/log/elasticsearch  bootstrap.memory\_lock: true  network.host: 192.168.91.55,127.0.0.1  http.port: 9200  http.cors.enabled: true  http.cors.allow-origin: "\*"  discovery.zen.ping.unicast.hosts: ["192.168.91.55", "192.168.91.56"]  discovery.zen.minimum\_master\_nodes: 2 |

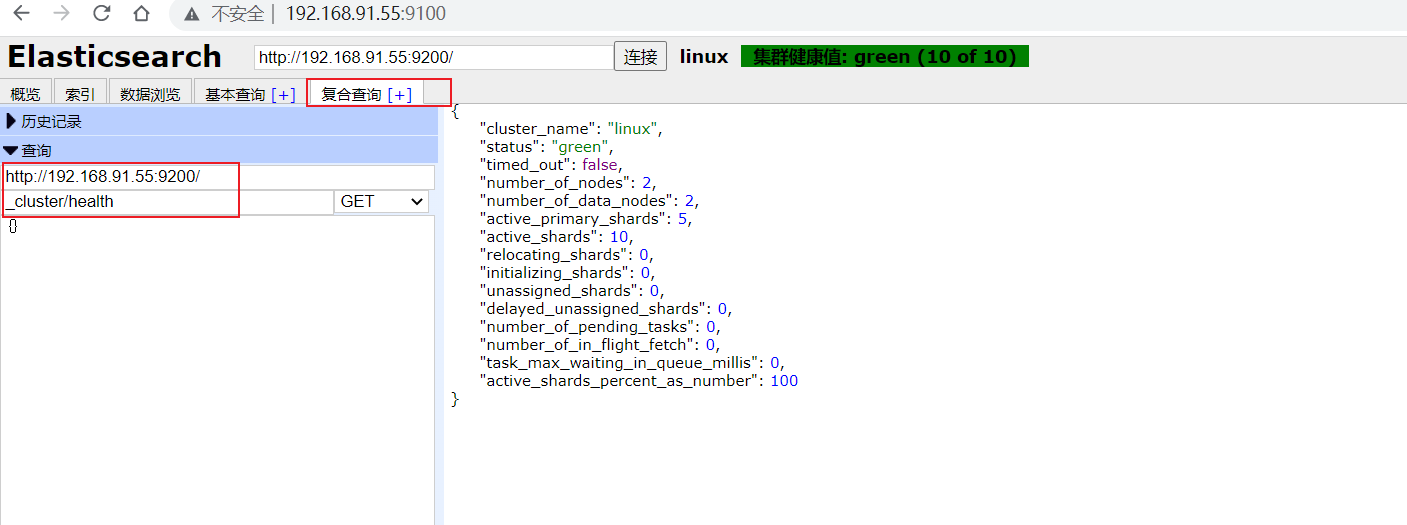
Node-2

|  |
| --- |
| cluster.name: linux  node.name: node-2  path.data: /data/elasticsearch  path.logs: /var/log/elasticsearch  bootstrap.memory\_lock: true  network.host: 192.168.91.56,127.0.0.1  http.port: 9200  http.cors.enabled: true  http.cors.allow-origin: "\*"  discovery.zen.ping.unicast.hosts: ["192.168.91.55", "192.168.91.56"]  discovery.zen.minimum\_master\_nodes: 2 |

Elasticsearch 集群-查看集群信息

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





status 字段是我们最关心的。

green 所有的主分片和副本分片都正常运行。

yellow 所有的主分片都正常运行，但不是所有的副本分片都正常运行。

red 有主分片没能正常运行。



**1.9 增加第三台节点**



|  |
| --- |
| cluster.name: linux  node.name: node-3  path.data: /data/elasticsearch  path.logs: /var/log/elasticsearch  bootstrap.memory\_lock: true  network.host: 192.168.91.57,127.0.0.1  http.port: 9200  http.cors.enabled: true  http.cors.allow-origin: "\*"  discovery.zen.ping.unicast.hosts: ["192.168.91.55", "192.168.91.57"]  discovery.zen.minimum\_master\_nodes: 2 |

**公式 master/2 +1; 由于3/2 + 1=2.5 不超过3，所以三个节点时还是配置discovery.zen.minimum\_master\_nodes为2**

查看集群节点：

curl -XGET 'http://127.0.0.1:9200/\_cat/nodes?pretty'

查看集群健康状态：

# curl -i -XGET <http://127.0.0.1:9200/_cluster/health?pretty>

**2、使用docker部署elasticsearch-head**

docker pull alivv/elasticsearch-head

docker run --name es-head -p 9100:9100 -dit alivv/elasticsearch-head

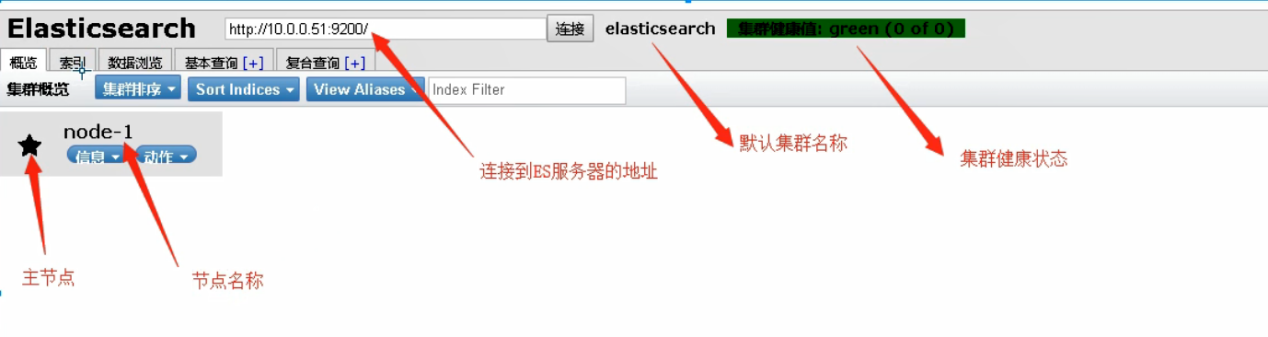
**修改ES配置文件支持跨域**

cat >> /etc/elasticsearch/elasticsearch.yml << EOF

http.cors.enabled: true

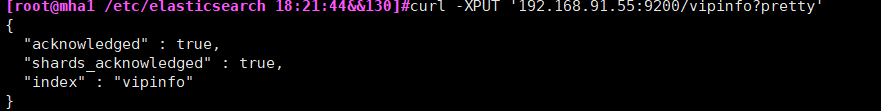
http.cors.allow-origin: "\*"

EOF



**3、创建索引（可有可无 作为了解下）**

curl -XPUT '192.168.91.55:9200/vipinfo?pretty'





插入文档数据

|  |
| --- |
| curl -XPUT '192.168.91.55:9200/vipinfo/user/1?pretty' -H 'Content-Type: application/json' -d'  {  "first\_name" : "John",  "last\_name": "Smith",  "age" : 25,  "about" : "I love to go rock climbing", "interests": [ "sports", "music" ]  }  ' |







|  |
| --- |
| curl -XPUT 'localhost:9200/vipinfo/user/2?pretty' -H 'Content-Type: application/json' -d' {  "first\_name": "Jane",  "last\_name" : "Smith",  "age" : 32,  "about" : "I like to collect rock albums", "interests": [ "music" ]  }' |

|  |
| --- |
| curl -XPUT 'localhost:9200/vipinfo/user/3?pretty' -H 'Content-Type: application/json' -d' {  "first\_name": "Douglas", "last\_name" : "Fir",  "age" : 35,  "about": "I like to build cabinets", "interests": [ "forestry" ]  }' |



查询索引中所有的

curl -XGET localhost:9200/vipinfo/user/\_search?pretty

查询指定文档数据

curl -XGET 'localhost:9200/vipinfo/user/1?pretty'

curl -XGET 'localhost:9200/vipinfo/user/2?pretty'

按条件查询文档数据

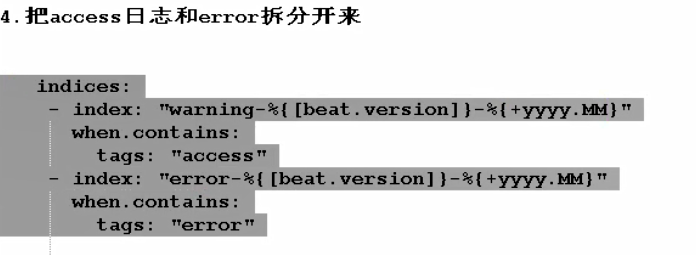
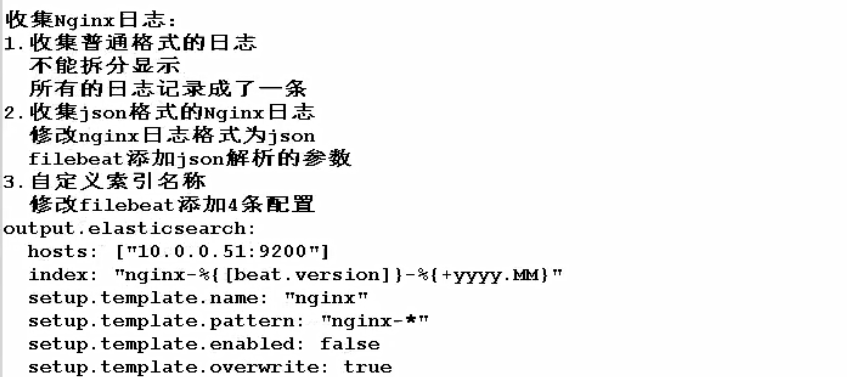
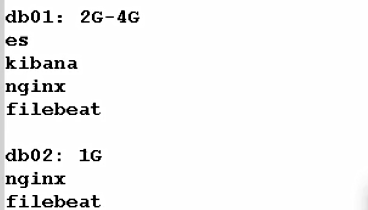
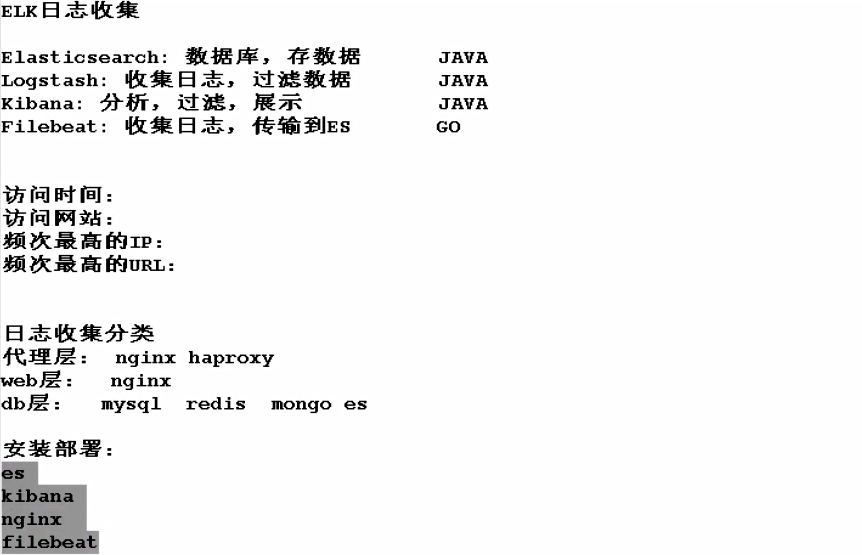
查询索引中符合条件的数据:搜索姓氏为Smith的雇员

curl -XGET 'localhost:9200/vipinfo/user/\_search?q=last\_name:Smith&pretty'

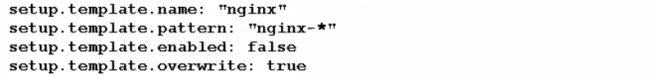
使用Query-string查询

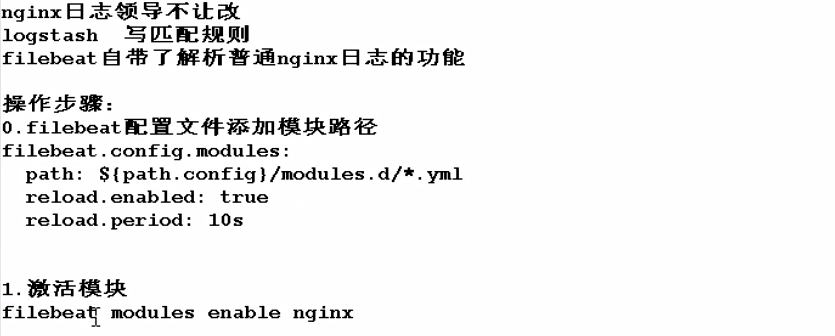
|  |
| --- |
| curl -XGET 'localhost:9200/vipinfo/user/\_search?pretty' -H 'Content-Type: application/json' -d'  {  "query" : {  "match" : {  "last\_name" : "Smith"  }  }  }  ' |

1. **ELK日志收集**



**以下配置只需要配置过使用自定义，就无需再另配。比如当要添加配置tomca日志时。同样对tomcat有效。（当某一台服务器集邮nginx又有tomcat时）**



1.命令行输入激活模块

[root@db01 ~]# filebeat modules enable nginx

2.修改nginx模块配置文件

[root@db01 ~]# cat /etc/filebeat/modules.d/nginx.yml |egrep -v "#|^$"

- module: nginx

access:

enabled: true

var.paths: ["/var/log/nginx/access.log"]

error:

enabled: true

var.paths: ["/var/log/nginx/error.log"]

3.修改nginx日志为普通格式

>/var/log/nginx/access.log

nginx -t

systemctl restart nginx

4.安装es2个插件

/usr/share/elasticsearch/bin/elasticsearch-plugin install file:///root/ingest-user-agent-6.6.0.zip

/usr/share/elasticsearch/bin/elasticsearch-plugin install file:///root/ingest-geoip-6.6.0.zip

5.重启es

systemctl restart elasticsearch

6.重启filebeat

systemctl restart filebeat

**4.2使用redis缓存配置优化，由于根据tags标签来区别的 ，output就不需要来判断了。**

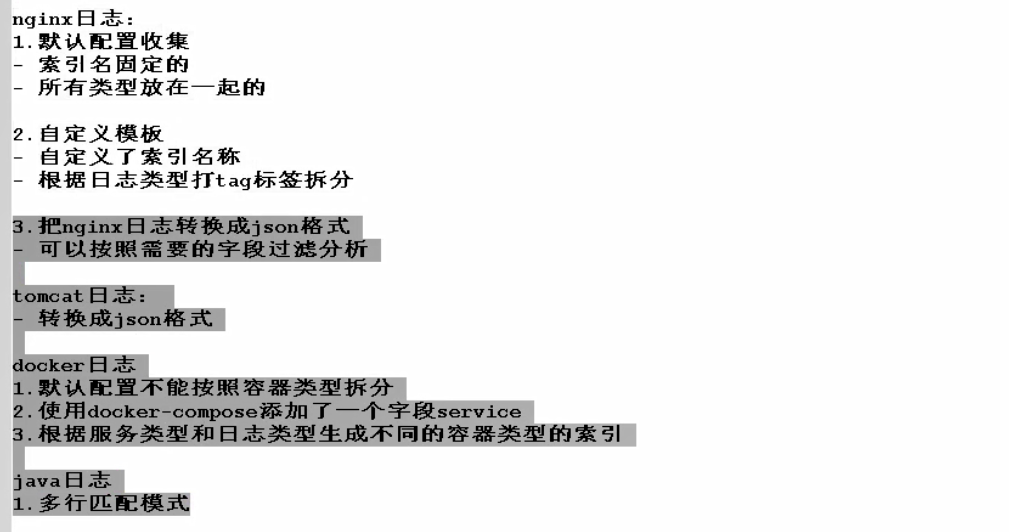
/etc/filebeat/filebeat.yml

|  |
| --- |
| filebeat.inputs:  - type: log  enabled: true  paths:  - /var/log/nginx/access.log  json.keys\_under\_root: true  json.overwrite\_keys: true  tags: ["access"]  - type: log  enabled: true  paths:  - /var/log/nginx/error.log  tags: ["error"]  output.redis:  hosts: ["192.168.91.55"]  keys:  - key: "filebeat" |

/etc/logstash/conf.d/redis.conf

|  |
| --- |
| input {  redis {  host => "192.168.91.55"  port => "6379"  db => "0"  key => "filebeat"  data\_type => "list"  }  }  filter {  mutate {  convert => ["upstream\_time", "float"]  convert => ["request\_time", "float"]  }  }  output {  stdout {}  if "access" in [tags] {  elasticsearch {  hosts => "http://192.168.91.55:9200"  manage\_template => false  index => "nginx\_access-%{+yyyy.MM}"  }  }  if "error" in [tags] {  elasticsearch {  hosts => "http://192.168.91.55:9200"  manage\_template => false  index => "nginx\_error-%{+yyyy.MM}"  }  }  } |

ELK知识回顾



# **收集nginx日志**

6版本使用beat.version beat.version是不能更改的

nginx-access-%{[beat.version]}-%{+yyyy.MM}

7版本使用agent.version

nginx-access-%{[agent.version]}-%{+yyyy.MM}

**索引删除的时候有两个地方要注意**

**一个是es集群的索引 一个是kibana的索引**

|  |
| --- |
| filebeat.inputs:  - type: log    enabled: true    paths:      - /var/log/nginx/access.log    json.keys\_under\_root: true    json.overwrite\_keys: true    tags: ["access"]  - type: log    enabled: true    paths:      - /var/log/nginx/error.log    tags: ["error"]  setup.kibana:    host: "10.0.0.51:5601"  output.elasticsearch:    hosts: ["10.0.0.51:9200"]  *#index: "nginx-%{[beat.version]}-%{+yyyy.MM}"*    indices:      - index: "nginx-access-%{[beat.version]}-%{+yyyy.MM}"        when.contains:          tags: "access"      - index: "nginx-error-%{[beat.version]}-%{+yyyy.MM}"        when.contains:          tags: "error"  setup.template.name: "nginx"  setup.template.pattern: "nginx-\*"  setup.template.enabled: false  setup.template.overwrite: true |

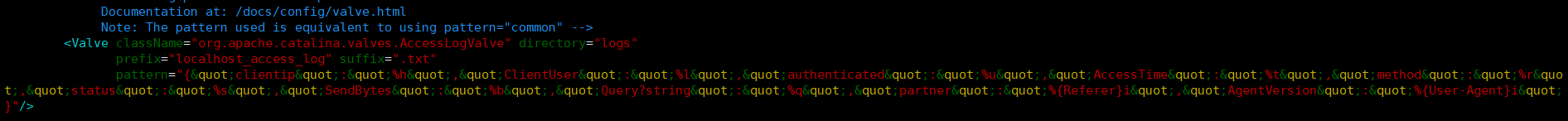
****

# **收集tomcat日志**

**vim ../conf/server.xml**

**添加以下日志格式**

|  |
| --- |
| pattern="{&quot;clientip&quot;:&quot;%h&quot;,&quot;ClientUser&quot;:&quot;%l&quot;,&quot;authenticated&quot;:&quot;%u&quot;,&quot;AccessTime&quot;:&quot;%t&quot;,&quot;method&quot;:&quot;%r&quot;,&quot;status&quot;:&quot;%s&quot;,&quot;SendBytes&quot;:&quot;%b&quot;,&quot;Query?string&quot;:&quot;%q&quot;,&quot;partner&quot;:&quot;%{Referer}i&quot;,&quot;AgentVersion&quot;:&quot;%{User-Agent}i&quot;}"/> |



vim filebeat.yaml

|  |
| --- |
| filebeat.inputs:  - type: log  enabled: true  paths:  - /home/java/wm/logs/localhost\_access\_log.\*.txt  json.keys\_under\_root: true  json.overwrite\_keys: true  tags: ["tomcat"]  setup.kibana:    host: "192.168.31.31:5601"  output.elasticsearch:  hosts: ["192.168.31.31:9200"]  indices:  - index: "tomcat-access-%{[beat.version]}-%{+yyyy.MM}"  when.contains:  tags: "tomcat"  setup.template.name: "nginx"  setup.template.pattern: "nginx-\*"  setup.template.enabled: false  setup.template.overwrite: true |



**setup.template.name: "nginx"**

**setup.template.pattern: "nginx-\*"**

**setup.template.enabled: false**

**setup.template.overwrite: true**

**看上去tomca和上面定义的nginx不匹配，其实第一次使用了自定义日志，后面就会用自定义的，不会再用原来的。**

# **收集java多行日志**



# **收集catalina.out日志**

**vim /etc/filebeat/filebeat.yml**

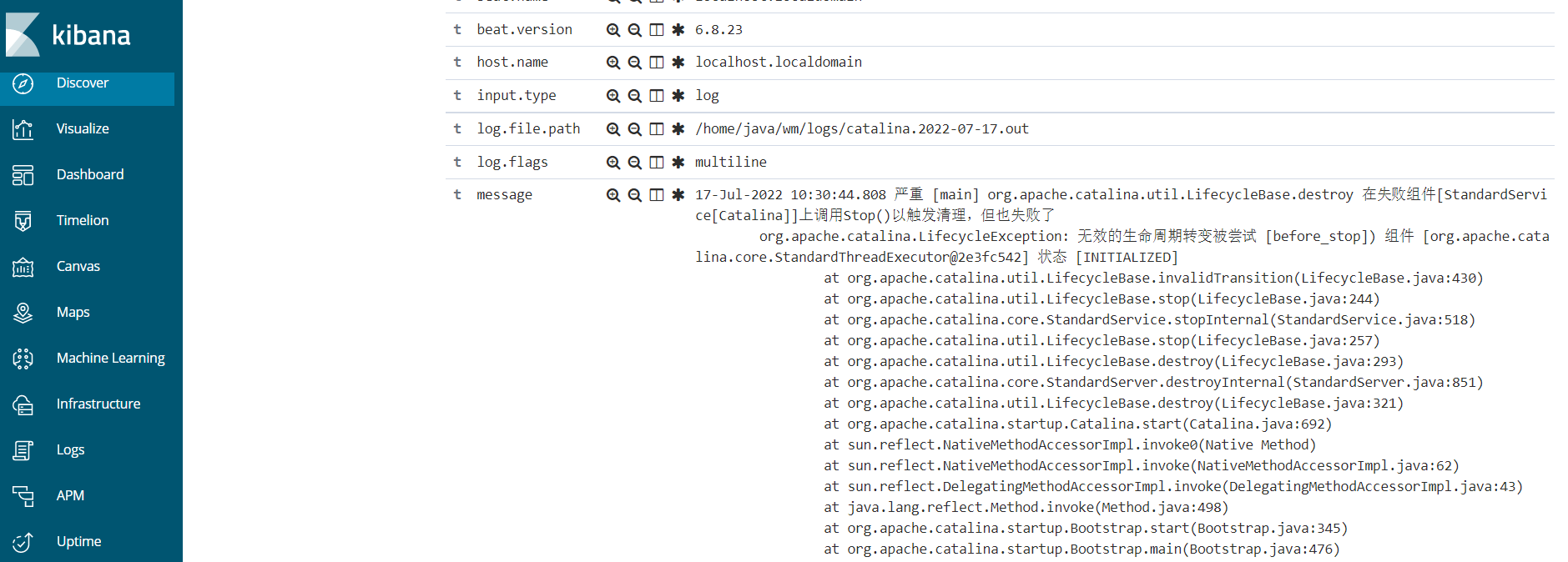
|  |
| --- |
| filebeat.prospectors:  - type: log  paths:  - /home/java/wm/logs/catalina.2022-07-17.out  fields:  app: www  type: tomcat-catalina  tags: ["catalina"]  fields\_under\_root: true  multiline:  #pattern: '^\['  pattern: '^[0-9]'  negate: true  match: after  output.redis:  hosts: ["192.168.31.31"]  password: "123.com"  key: "filebeat"  db: 0  datatype: list |

**vim /etc/logstash/conf.d/tomcat.conf**

|  |
| --- |
| input {  redis {  host => "192.168.31.31"  port => "6379"  password => "123.com"  db => "0"  key => "filebeat"  data\_type => "list"  }  }  filter {  }  output {  stdout {}  if "tomcat\_catalina" in [tags] {  elasticsearch {  hosts => "http://192.168.31.31:9200"  manage\_template => false  index => "tomcat\_catalina-%{+yyyy.MM.dd}"  }  }  } |

**验证**

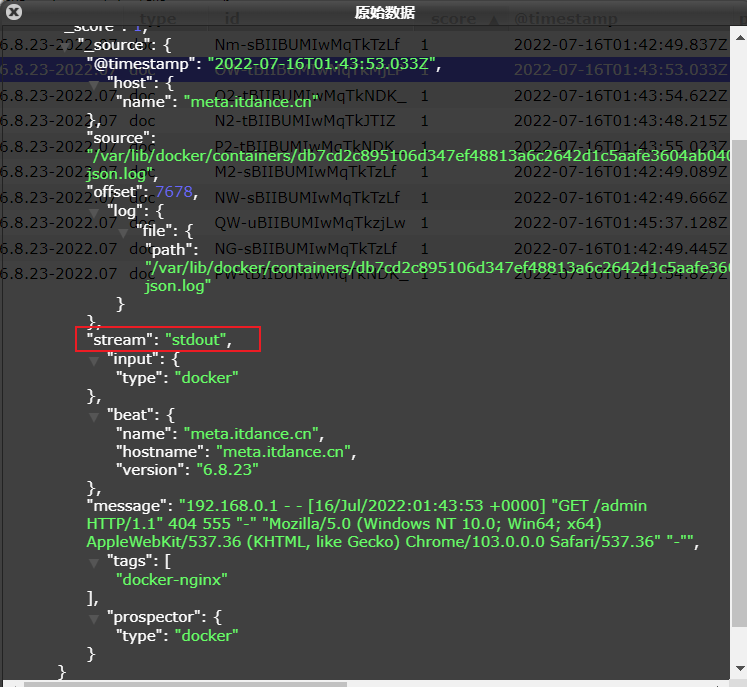
/usr/share/logstash/bin/logstash -f /etc/logstash/conf.d/tomcat.conf



# **收集docker日志**

**单个容器**

|  |
| --- |
| filebeat.inputs:  - type: docker  containers.ids:  - 'db7cd2c895106d347ef48813a6c2642d1c5aafe3604ab040a91b594552ca1251'  tags: ["docker-nginx"]  output.elasticsearch:  hosts: ["localhost:9200"]  indices:  - index: "docker-access-%{[beat.version]}-%{+yyyy.MM}"  when.contains:  stream: "stdout"  - index: "docker-error-%{[beat.version]}-%{+yyyy.MM}"  when.contains:  stream: "stderr"  setup.template.name: "docker"  setup.template.pattern: "docker-\*"  setup.template.enabled: false  setup.template.overwrite: true |



**多个容器**

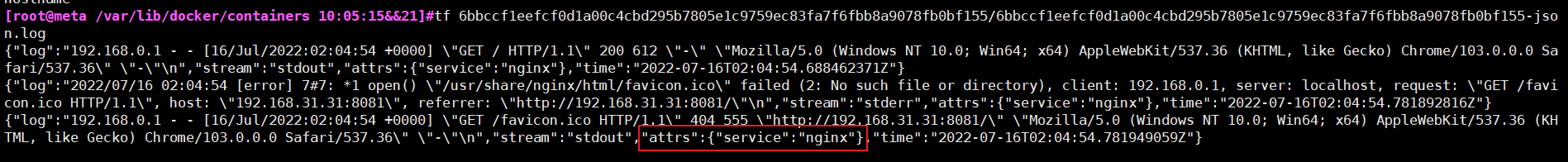
编写docker-compose.yml

|  |
| --- |
| **version: '3'**  **services:**  **nginx:**  **image: nginx:1.15.8**  **labels:**  **service: nginx**  **logging:**  **options:**  **labels: "service"**  **ports:**  **- "8081:80"**  **db:**  **image: nginx:latest**  **labels:**  **service: db**  **logging:**  **options:**  **labels: "service"**  **ports:**  **- "8082:80"** |

运行docker-compose.yml

docker-compose up -d

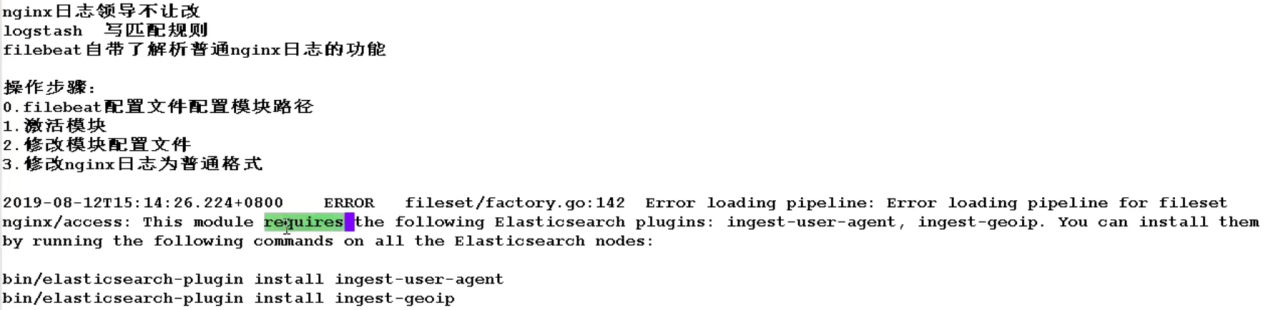
检查日志是否增加了lable标签



配置filebeat通过服务类型和日志类型多条件创建不同索引

|  |
| --- |
| filebeat.inputs:  - type: log  enabled: true  paths:  - /var/lib/docker/containers/\*/\*-json.log  json.keys\_under\_root: true  json.overwrite\_keys: true  output.elasticsearch:  hosts: ["192.168.47.175:9200"]  indices:  - index: "docker-nginx-access-%{[beat.version]}-%{+yyyy.MM.dd}"  when.contains:  attrs.service: "nginx"  stream: "stdout"  - index: "docker-nginx-error-%{[beat.version]}-%{+yyyy.MM.dd}"  when.contains:  attrs.service: "nginx"  stream: "stderr"  - index: "docker-db-access-%{[beat.version]}-%{+yyyy.MM.dd}"  when.contains:  attrs.service: "db"  stream: "stdout"  - index: "docker-db-error-%{[beat.version]}-%{+yyyy.MM.dd}"  when.contains:  attrs.service: "db"  stream: "stderr"  setup.template.name: "docker"  setup.template.pattern: "docker-\*"  setup.template.enabled: false  setup.template.overwrite: true |

# **使用模版配置nginx正常日志**

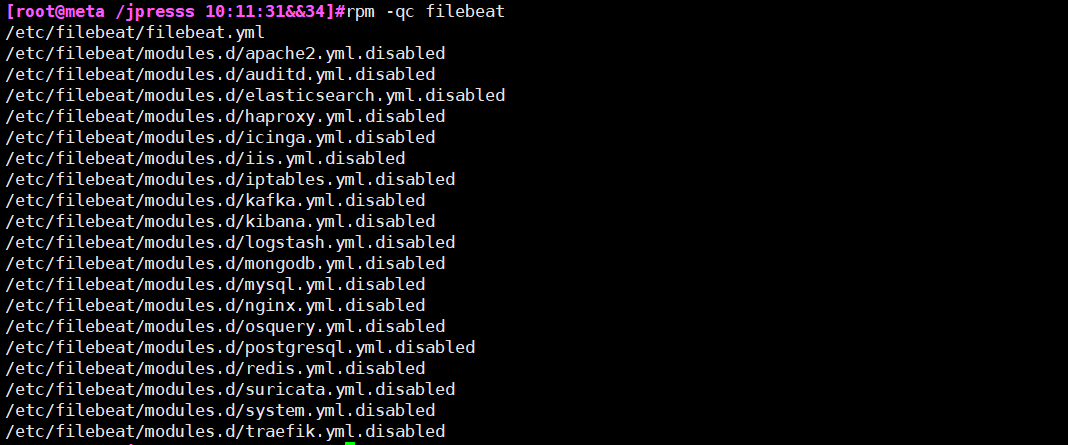


|  |
| --- |
| **1.命令行输入激活模块**  **[root@db01 ~]# filebeat modules enable nginx**  **2.修改nginx模块配置文件**  **[root@db01 ~]# cat /etc/filebeat/modules.d/nginx.yml |egrep -v "#|^$"**  **- module: nginx**  **access:**  **enabled: true**  **var.paths: ["/var/log/nginx/access.log"]**  **error:**  **enabled: true**  **var.paths: ["/var/log/nginx/error.log"]**  **3.修改nginx日志为普通格式**  **>/var/log/nginx/access.log**  **nginx -t**  **systemctl restart nginx**  **4.重启es**  **systemctl restart elasticsearch** |

6.7之后这两个插件默认集成到了elasticsearch，不需要单独安装了

**社区论坛：**

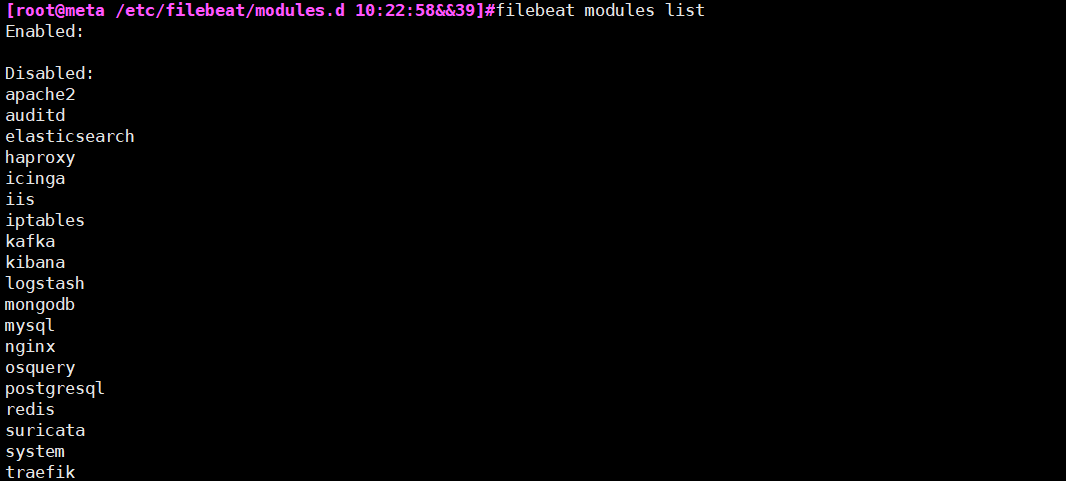
<https://discuss.elastic.co/t/filebeat-module-custom-index/181350>



**此时查看是都没有激活的 disabled状态**

**查看支持的模块**

filebeat modules list

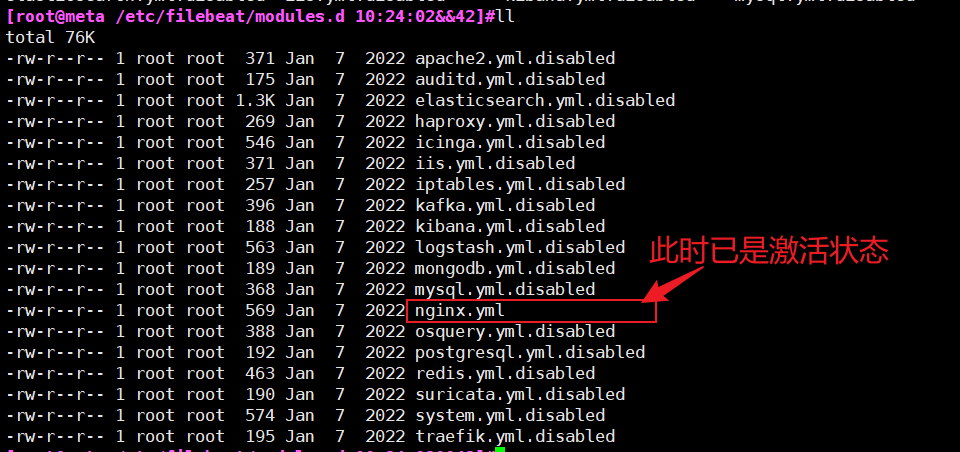


**1、配置filebea.yml**

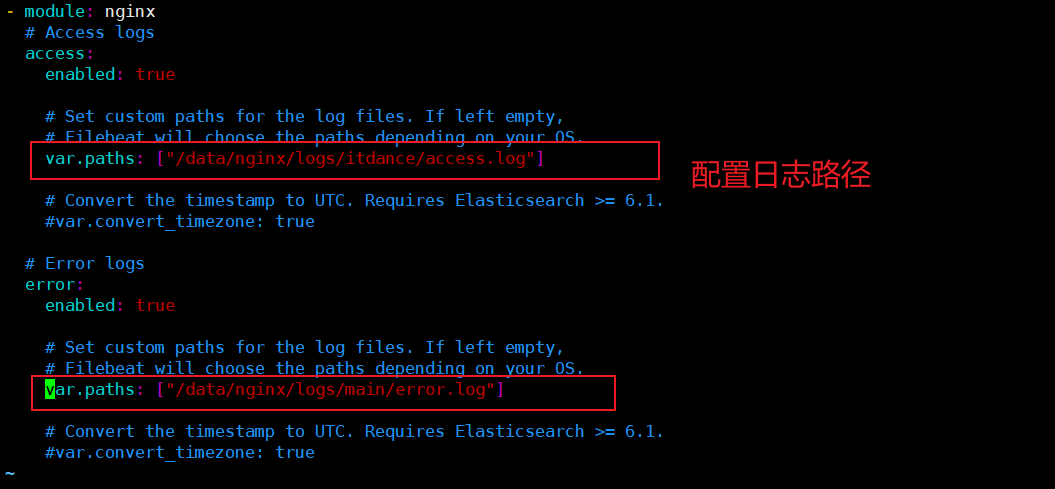
|  |
| --- |
| filebeat.config.modules:    path: ${path.config}/modules.d/\*.yml #配置使用模块的路径    reload.enabled: true    reload.period: 10s  setup.kibana:    host: "10.0.0.51:5601"    output.elasticsearch:    hosts: ["10.0.0.51:9200"]    indices:    - index: "nginx-access-%{[beat.version]}-%{+yyyy.MM}"      when.contains:        fileset.name: "access"    - index: "nginx-error-%{[beat.version]}-%{+yyyy.MM}"      when.contains:        fileset.name: "error"  setup.template.name: "nginx"  setup.template.pattern: "nginx-\*"  setup.template.enabled: false  setup.template.overwrite: true |

**2、激活模块**

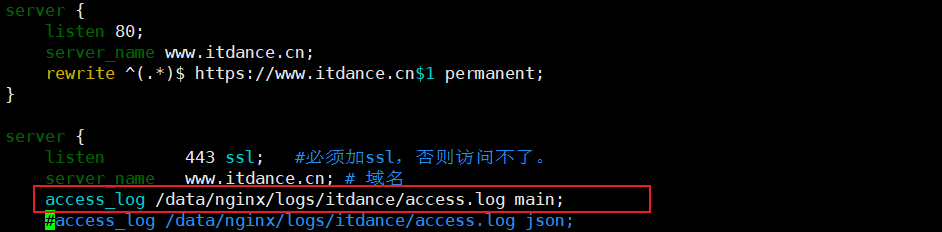
filebeat modules enable nginx



**3、配置日志路径**



**4、修改nginx日志格式为普通格式**



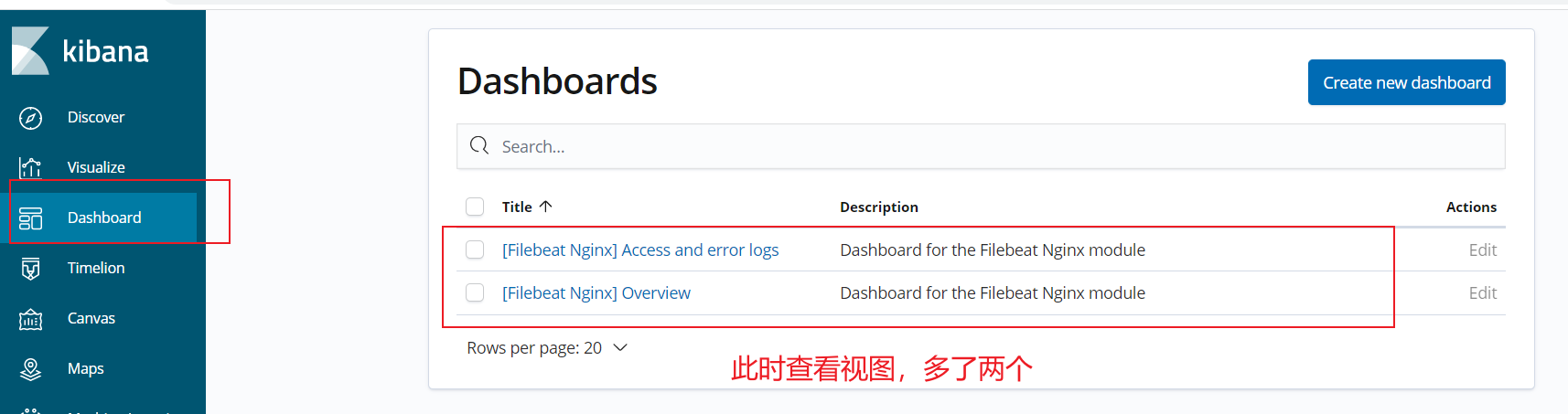
清空日志

> /data/nginx/logs/itdance/access.log

filebeat setup 会自动把图导入到kibana

1. **重启filebeat**

|  |
| --- |
| cp -a /usr/share/filebeat/kibana /root  cd /root/kibana/ && rm -rf 5  cd /root/kibana/6/dashboard && find . -type f ! -name "\*nginx\*"|xargs rm -rf  cd /root/kibana/6/dashboard && rm -f ml-nginx-\*    cd /root/kibana/6/dashboard  sed -i 's#filebeat\-\\*#nginx\-\\*#g' Filebeat-nginx-logs.json  sed -i 's#filebeat\-\\*#nginx\-\\*#g' Filebeat-nginx-overview.json #匹配nginx开头的  /root/kibana/6/index-pattern  sed -i 's#filebeat\-\\*#nginx\-\\*#g' filebeat.json  替换索引名称  filebeat setup --dashboards -E setup.dashboards.directory=/root/kibana/ |



# **使用单台redis作为缓存**

**Es只支持单台redis，不支持redis集群。可以考虑使用keeplive，upstream 将备用的redis标记为backup,也有一个缺陷，就是可能在down机的redis数据可能会丢失。**

### 安装启动测试redis

[root@elk-175 ~]# yum install redis -y

[root@elk-175 ~]# systemctl status redis

[root@elk-175 ~]# redis-cli

127.0.0.1:6379> set k1 v1

OK

127.0.0.1:6379> GEt k1

"v1"

127.0.0.1:6379>

### 配置nginx的json日志

将nginx的日志调整为json格式

log\_format json '{ "time\_local": "$time\_local", '

'"remote\_addr": "$remote\_addr", '

'"referer": "$http\_referer", '

'"request": "$request", '

'"status": $status, '

'"bytes": $body\_bytes\_sent, '

'"agent": "$http\_user\_agent", '

'"x\_forwarded": "$http\_x\_forwarded\_for", '

'"up\_addr": "$upstream\_addr",'

'"up\_host": "$upstream\_http\_host",'

'"upstream\_time": "$upstream\_response\_time",'

'"request\_time": "$request\_time"' ' }';

### 配置filebeat写入到一个key中

|  |
| --- |
| filebeat.inputs:  - type: log    enabled: true    paths:      - /var/log/nginx/access.log    json.keys\_under\_root: true    json.overwrite\_keys: true    tags: ["access"]  - type: log    enabled: true    paths:      - /var/log/nginx/error.log    tags: ["error"]  setup.kibana:    host: "10.0.0.51:5601"  output.redis:    hosts: ["10.0.0.51"]    key: "filebeat" |



### logstash根据tag区分一个key里的不同日志

[root@elk-175 ~]# cat /etc/logstash/conf.d/redis.conf

|  |
| --- |
| input {  redis {  host => "10.0.0.51"  port => "6379"  db => "0"  key => "filebeat"  data\_type => "list"  }  }  filter {  mutate {  convert => ["upstream\_time", "float"]  convert => ["request\_time", "float"]  }  }  output {  stdout {}  if "access" in [tags] {  elasticsearch {  hosts => "http://localhost:9200"  manage\_template => false  index => "nginx\_access-%{+yyyy.MM.dd}"  }  }  if "error" in [tags] {  elasticsearch {  hosts => "http://localhost:9200"  manage\_template => false  index => "nginx\_error-%{+yyyy.MM.dd}"  }  }  } |

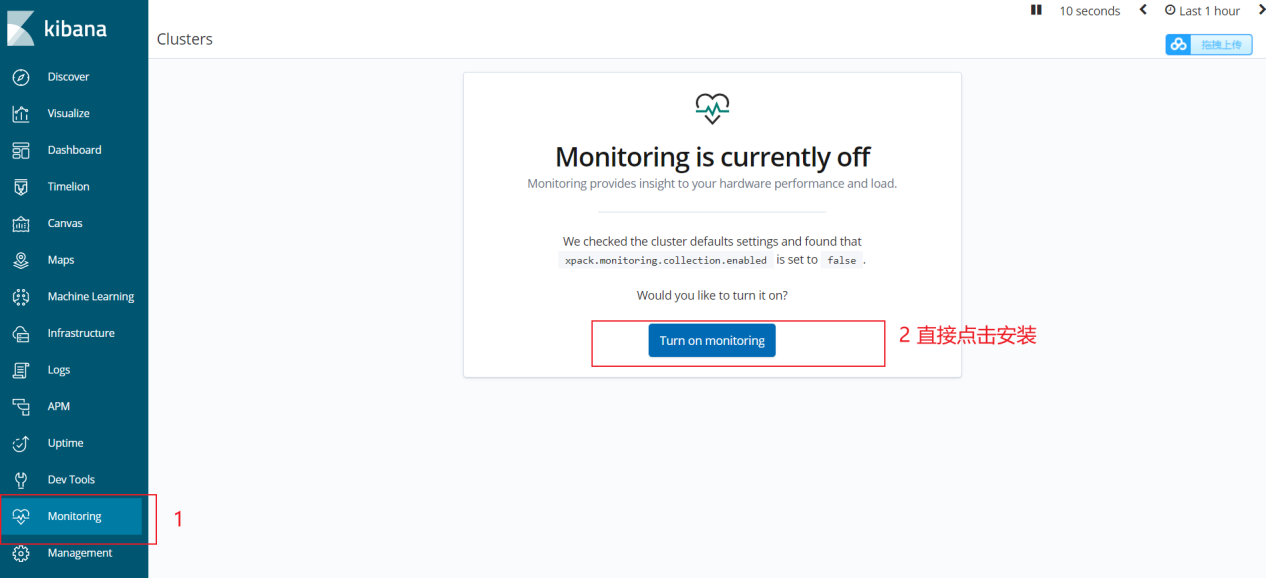


**前台启动**

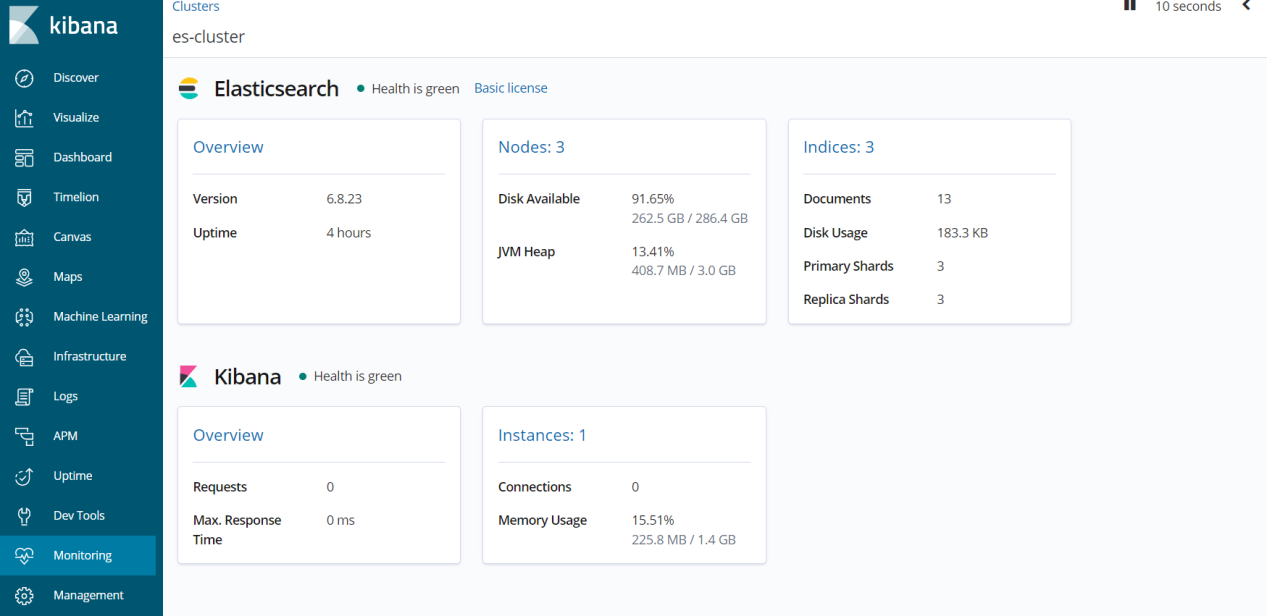
/usr/share/logstash/bin/logstash -f /etc/logstash/conf.d/redis.conf 前台启动，可以查看过程

systemctl start logstash

# **配置监控es集群**



结果如下：



**其他**

filebeat.yml中fields和fields\_under\_root: tru是一起出现的

|  |
| --- |
| fields:      app: www      type: nginx-access    fields\_under\_root: true |

filebeat.yml中将message字段中的键值抽离出来，不至于放在同一个键里面

|  |
| --- |
| json.keys\_under\_root: true  json.overwrite\_keys: true |