

# ELK 环境搭建和使用

Name : 曲中岭  
Email: [zlingqu@126.com](mailto:zlingqu@126.com)  
Q Q : 441869115

## 第一章 部署准备

### 1.1 目的

使用 ELK 搭建日志收集和分析系统, 将所有的应用日志、系统日志等做统一收集、存储、查询、分析等管理动作, 提供 API 和 web 供研发、运维、运营等各自获取自己关心的内容。

### 1.2 规划

IP	172.16.6.11	172.16.7.46
OS	CentOS 7.5 x64	Ubuntu 14.04 x64
Elasticsearch	√	
Logstash		√
Kibana	√	
jdk	1.8.0_191	1.8.0_91
jdk-mode	openjdk by rpm	oracle-jdk by env
Running mode	systemd	upstart(service)
nginx		√

### 1.3 配置 java 环境

我这里选择最简单的方法

```
yum install -y java  
java -version
```

```

已安装:
java-1.8.0-openjdk.x86_64 1:1.8.0.191.b12-1.el7_6

作为依赖被安装:
copy-jdk-configs.noarch 0:3.3-10.el7_5      giflib.x86_64 0:4.1.6-9.el7      java-1
libICE.x86_64 0:1.0.9-9.el7                  libSM.x86_64 0:1.2.2-2.el7      libX11
libXau.x86_64 0:1.0.8-2.1.el7                 libXcomposite.x86_64 0:0.4.4-4.1.el7  libXext
libXrender.x86_64 0:0.9.10-1.el7              libXtst.x86_64 0:1.2.3-1.el7    libfont
libpng.x86_64 2:1.5.13-7.el7_2               libxcb.x86_64 0:1.13-1.el7      lkscpt
python-lxml.x86_64 0:3.2.1-4.el7              ttmkfdirdir.x86_64 0:3.0.9-42.el7  tzdata
xorg-x11-fonts-Type1.noarch 0:7.5-9.el7

作为依赖被升级:
nspr.x86_64 0:4.19.0-1.el7_5      nss.x86_64 0:3.36.0-7.el7_5      nss-softokn.x86_64 0
nss-tools.x86_64 0:3.36.0-7.el7_5  nss-util.x86_64 0:3.36.0-1.el7_5

完毕!
[root@tidb1 ~]# java -version
openjdk version "1.8.0_191"
OpenJDK Runtime Environment (build 1.8.0_191-b12)
OpenJDK 64-Bit Server VM (build 25.191-b12, mixed mode)

```

如果，自动安装 openjdk，版本是 1.8.0\_191

## 1.4 修改系统参数

### 1.4.1 文件描述符

同时运行程序的用户的如下命令输出的值不小于 65536

```
ulimit -n
```

可在/etc/security/limits.conf 中配置以下两行:

```

*          soft          nofile          65536
*          hard          nofile          65536

```

程序启动后，可使用如下命令进行确认:

```

curl -X GET http://127.0.0.1:9200/_nodes/stats/process?filter_path=**.max_file_descriptors
curl -X GET http://10.28.93.179:9200/_nodes/stats/process?pretty|grep max_file_descriptors

```

```

root@iZuf652c004lmvft5cjo9nZ:~# curl -X GET http://10.28.93.179:9200/_nodes/stats/process?pretty|grep max_file_descriptors
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload    Total   Spent    Left   Speed
100 2481 100 2481    0     0  419k      0 --:--:-- --:--:-- --:--:-- 484k
      "max_file_descriptors" : 65536,
      "max_file_descriptors" : 65536,
      "max_file_descriptors" : 65536,
root@iZuf652c004lmvft5cjo9nZ:~#

```

### 1.4.2 虚拟内存

```
echo "vm.max_map_count = 262144" >> /etc/sysctl.conf
```

```
sysctl -p
```

```
sysctl -a|grep vm.max
```

如果使用 rpm 或者 deb 包安装的，这一步会自动配置。

### 1.4.3 最大线程数

保证不小于 4096

```
ulimit -u 4096
```

也可以在 limits.conf 文件中进行配置

```
*      soft    nproc    65535
*      hard    nproc    65535
```

如果使用 rpm 或者 deb 包安装的，并且是使用 systemd 管理的，这一步会自动配置。

### 1.4.4 禁用交换内存

#### 方法 1:

临时关闭 swap

```
swapoff -a
```

并修改/etc/fstab 中的包含 swap 挂载的行。

#### 方法 2:

启用配置文件/etc/elasticsearch/elasticsearch.yml 中的如下配置项，默认就是 true

```
bootstrap.memory_lock: true
```

## 第二章 Elasticsearch 部署

官方文档:

<https://www.elastic.co/guide/en/elasticsearch/reference/current/index.html>

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

由于机器不够，这里使用单机模式部署。

### 2.1 简单介绍

传统关系型数据库和 es 的对比:

Relational database		Elasticsearch	
Databases	数据库	Index	索引
Table	表	Type	类型
Row	行	Document	文档
Column	列	Field	域 (字段)
Schema	数据类型	Mapping	数据类型
Index	索引, 例如 B-Tree	Everything is index	一切皆索引, 例如倒排索引(inverted index)
acid	事务	————	不支持,也不支持修改
Update	修改	不支持	不支持
row storage	行存储	文档存储	文档存储
SQL	sql 语法	Query DSL	基于 JSON 的 query DSL 查询语言
SELECT *	查询	GET http://	get 请求
FROM			
UPDATE table	更新	PUT http://	put 请求, 重建索引
SET			
INSERT INTO table (*)	增加	POST http:// PUT http://	post 请求
DELETE FROM table	删除	DELETE http://	delete 请求
TCP/IP, Unix Socket, Share Memory 等	协议	http, thrift, servlet, memcached, zeroMQ 等	协议

比如要存储员工数据:

我们首先要做的是存储员工数据, 每个文档代表一个员工。在 Elasticsearch 中存储数据的行为就叫做索引(indexing)。

Elasticsearch 集群可以包含多个索引(indices) (数据库), 每一个索引可以包含多个类型(types) (表), 每一个类型包含多个文档(documents) (行), 然后每个文档包含多个字段(Fields) (列)。

可以使用如下语句添加数据:

```
curl -X PUT http://127.0.0.1:9200/megacorp/employee/1 \
-H 'Content-Type: application/json' \
-d '
{
  "first_name" : "John",
  "last_name" : "Smith",
  "age" :      25,
  "about" :    "I love to go rock climbing",
  "interests": [ "sports", "music" ]
}'
```

我们看到 path:/megacorp/employee/1 包含三部分信息:

名字	说明
megacorp	索引名
employee	类型名
1	这个员工的 ID

请求实体 (JSON 文档), 包含了这个员工的所有信息。他的名字叫“John Smith”, 25 岁, 喜欢攀岩。

很简单吧! 它不需要你做额外的管理操作, 比如创建索引或者定义每个字段的数据类型。我们能够直接索引文档, Elasticsearch 已经内置所有的缺省设置, 所有管理操作都是透明的。使用如下语句获取数据:

```
curl http://127.0.0.1:9200/megacorp/employee/_search
curl http://127.0.0.1:9200/megacorp/employee/_search?pretty
```

其他各种搜索方式见:

[https://es.xiaoleilu.com/010\\_Intro/30\\_Tutorial\\_Search.html](https://es.xiaoleilu.com/010_Intro/30_Tutorial_Search.html)

```
[root@tidb1 elasticsearch]# curl -X PUT http://127.0.0.1:9200/megacorp/employee/1 \
> -H 'Content-Type: application/json' \
> -d '
> {
>   "first_name" : "John",
>   "last_name" : "Smith",
>   "age" :      25,
>   "about" :    "I love to go rock climbing",
>   "interests": [ "sports", "music" ]
> }'
{"_index":"megacorp","_type":"employee","_id":"1","_version":1,"result":"created","_shards":{"total":2,"successful":2,"skipped":0,"failed":0}}
[root@tidb1 elasticsearch]#
[root@tidb1 elasticsearch]#
[root@tidb1 elasticsearch]# curl http://127.0.0.1:9200/megacorp/employee/_search
{"took":85,"timed_out":false,"_shards":{"total":5,"successful":5,"skipped":0,"failed":0},"hits":{"total":1,"max_score":1,"hits":[{"_source":{"first_name" : "John",
"last_name" : "Smith",
"age" :      25,
"about" :    "I love to go rock climbing",
"interests": [ "sports", "music" ]
}}]}}[root@tidb1 elasticsearch]#
```

## 2.2 下载

下载地址：

<https://www.elastic.co/downloads/elasticsearch#ga-release>

建议直接使用 deb 或者 rpm 包，便于升级。

例如：

wget <https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-6.5.2.rpm>

## 2.3 安装

安装、运行、添加到开机启动项。

```
rpm -ivh elasticsearch-6.5.2.rpm
```

或者

```
rpm -ivh https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-6.5.2.rpm
```

该文件 150M 左右

```
systemctl daemon-reload
systemctl enable elasticsearch
systemctl start elasticsearch
systemctl status elasticsearch
```

## 2.4 配置文件

### 2.4.1 配置文件

配置文件路径：/etc/elasticsearch

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

其中配置了，data、log 路径等，可根据需要进行修改。

### 2.4.2 网络监听

默认情况下，Elasticsearch 假定您正在开发模式下工作。如果未正确配置上述任何设置，则会向日志文件写入警告，但您将能够启动并运行 Elasticsearch 节点。

一旦配置了类似的网络设置 network.host，Elasticsearch 就会假定您正在转向生产并将上述警告升级为异常。这些异常将阻止您的 Elasticsearch 节点启动。这是一项重要的安全措施，可确保您不会因服务器配置错误而丢失数据。

比如配置：

```
network.host: 0.0.0.0
```

### 2.4.3 集群名字

如下，应该配置一个有意义的名字，及时是单机情况下也应该修改这个配置，防止内网有有相同的集群产生冲突

```
cluster.name: quzl
```

### 2.4.4 可用内存

jvm.options 中进行配置如下：

```
-Xms1g  
-Xmx1g
```

其中 Xmx 为不超过物理 RAM 的 50%

### 2.4.4 JAVA\_HOME

如果使用解压 jdk 的方式安装的 java 环境，需要在/etc/init.d/elasticsearch 文件中添加如下类似变量

```
JAVA_HOME=/usr/local/java
```

## 2.5 观察

- 1) 自动创建用户 elasticsearch 运行程序。
- 2) 启动 TCP9200、9300 端口
- 3) 版本是 6.5.2， lucene 版本是 7.5.0

```
[root@tidb1 ~]# id elasticsearch
uid=997(elasticsearch) gid=995(elasticsearch) 组=995(elasticsearch)
[root@tidb1 ~]#
[root@tidb1 ~]#
[root@tidb1 ~]# ps -ef|grep search
elastic+ 3472    1  S 14:13 ?        00:00:36 /bin/java -Xms1g -Xmx1g -XX:+UseConcMarkSweepGC
sslm -Djava.awt.headless=true -Dfile.encoding=UTF-8 -Djna.nosys=true -XX:-OmitStackTraceInFast
tyPerThread=0 -Dlog4j.shutdownHookEnabled=false -Dlog4j2.disable.jmx=true -Djava.io.tmpdir=/tm
-XX:ErrorFile=/var/log/elasticsearch/hs_err_pid%p.log -XX:+PrintGCDetails -XX:+PrintGCDateStam
ch/gc.log -XX:+UseGCLogFileRotation -XX:NumberOfGCLogFiles=32 -XX:GCLogFileSize=64m -Des.path.
-Des.distribution.type=rpm -cp /usr/share/elasticsearch/lib/* org.elasticsearch.bootstrap.Elas
elastic+ 3525  3472  0 14:13 ?        00:00:00 /usr/share/elasticsearch/modules/x-pack-ml/pla
root    3578  3146  0 14:25 pts/0    00:00:00 grep --color=auto search
[root@tidb1 ~]#
[root@tidb1 ~]#
[root@tidb1 ~]# netstat -tnlp|grep 3472
tcp6     0      0 127.0.0.1:9200        :::*           LISTEN      3472/java
tcp6     0      0 :::1:9200             :::*           LISTEN      3472/java
tcp6     0      0 127.0.0.1:9300        :::*           LISTEN      3472/java
tcp6     0      0 :::1:9300             :::*           LISTEN      3472/java
[root@tidb1 ~]#
[root@tidb1 ~]#
```

```
[root@tidb1 ~]# curl 127.0.0.1:9200
{
  "name" : "2WapIHv",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "wXa6TR-zTouVidxlr8GtsQ",
  "version" : {
    "number" : "6.5.2",
    "build_flavor" : "default",
    "build_type" : "rpm",
    "build_hash" : "9434bed",
    "build_date" : "2018-11-29T23:58:20.891072Z",
    "build_snapshot" : false,
    "lucene_version" : "7.5.0",
    "minimum_wire_compatibility_version" : "5.6.0",
    "minimum_index_compatibility_version" : "5.0.0"
  },
  "tagline" : "You Know, for Search"
}
```

## 2.6 插件管理

```
cd /usr/share/elasticsearch/bin
./elasticsearch-plugin -h
```

有如下参数：

list - Lists installed elasticsearch plugins

install - Install a plugin

remove - removes a plugin from Elasticsearch

-h, --help      show help

比如安装中文分词插件,版本要一致:

```
cd /usr/share/elasticsearch
./bin/elasticsearch-plugin install https://github.com/medcl/elasticsearch-analysis-ik/releases/download/v6.5.2/elasticsearch-analysis-ik-6.5.2.zip
```

```

root@iZuf652c0041mvft5cjo9mZ:/usr/share/elasticsearch# ./bin/elasticsearch-plugin install https://github.com/medcl/elasticsea
-6.5.2.zip
-> Downloading https://github.com/medcl/elasticsearch-analysis-ik/releases/download/v6.5.2/elasticsearch-analysis-ik-6.5.2.zip
[=====] 100%
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@      WARNING: plugin requires additional permissions      @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
* java.net.SocketPermission * connect,resolve
See http://docs.oracle.com/javase/8/docs/technotes/guides/security/permissions.html
for descriptions of what these permissions allow and the associated risks.

Continue with installation? [y/N]y
-> Installed analysis-ik
root@iZuf652c0041mvft5cjo9mZ:/usr/share/elasticsearch#
root@iZuf652c0041mvft5cjo9mZ:/usr/share/elasticsearch# ./bin/elasticsearch-plugin list
analysis-ik
root@iZuf652c0041mvft5cjo9mZ:/usr/share/elasticsearch#
root@iZuf652c0041mvft5cjo9mZ:/usr/share/elasticsearch# ls
bin  lib  modules  NOTICE.txt  plugins  README.textile
root@iZuf652c0041mvft5cjo9mZ:/usr/share/elasticsearch# ls plugins/
analysis-ik
root@iZuf652c0041mvft5cjo9mZ:/usr/share/elasticsearch#

```

### 验证中文分词效果

```
curl -XGET 10.28.93.179:9200/_analyze?pretty \
```



```
-H 'Content-Type: application/json' \
-d'
{
  "analyzer":"ik_max_word",
  "text":"中华人民共和国国歌"
}'
```

```
root@iZuf652c0o41mvft5cjo9mZ:~#
root@iZuf652c0o41mvft5cjo9mZ:~# curl -XGET 10.28.93.179:9200/_analyze -H 'Content-Type: application/json' -d'
{
  "analyzer":"ik_max_word",
  "text":"中华人民共和国国歌"
}'
root@iZuf652c0o41mvft5cjo9mZ:~# curl -XGET 10.28.93.179:9200/_analyze?pretty \
> -H 'Content-Type: application/json' \
> -d'
> {
>   "analyzer":"ik_max_word",
>   "text":"中华人民共和国国歌"
> }'
{
  "tokens" : [
    {
      "token" : "中华人民共和国",
      "start_offset" : 0,
      "end_offset" : 7,
      "type" : "CN_WORD",
      "position" : 0
    },
    {
      "token" : "中华人民",
      "start_offset" : 0,
      "end_offset" : 4,
      "type" : "CN_WORD"
```

## 2.7 es 语法

### 2.7.1 索引/数据库

查看健康状态:

```
curl -XGET '10.80.225.121:9200/_cat/health?v'
```

创建索引

```
curl -XPUT '10.80.225.121:9200/eshop_product_ent'
```

查看索引:

```
curl -XGET '10.80.225.121:9200/_cat/indices?v'
```

```
curl -XGET '10.80.225.121:9200/_cat/indices/eshop_product_ent?v'
```

删除索引

```
curl -XDELETE '10.80.225.121:9200/eshop_product_ent'
```

```
[root@esdb1 ~]# curl 127.0.0.1:9200/_cat/indices
yellow open eshop_product_ent 7800b8HtTfyFbBFv8ticYg 5 1 71185 12035 68.6mb 68.6mb
green open .monitoring-es-6-2019.03.06 3c6bIBNKtGaKjVLJ3EMGNA 1 0 202434 312 105.7mb 105.7mb
green open .monitoring-es-6-2019.03.01 dqSfDkF6T32lX_cNmPqDNA 1 0 700972 13540 348.5mb 348.5mb
green open .monitoring-kibana-6-2019.02.28 7nllsQqVQKK35d8tzxCGcQ 1 0 10290 0 1.6mb 1.6mb
green open .monitoring-es-6-2019.03.03 94e_x8kWQq6W5-EoyuZTEw 1 0 921828 11700 466.6mb 466.6mb
green open .monitoring-kibana-6-2019.03.05 BkTp-MTuRVKamzoagXLww 1 0 8636 0 2.3mb 2.3mb
green open .monitoring-kibana-6-2019.03.06 u2PbwmmQS3qkys3EYSbkug 1 0 8636 0 2.3mb 2.3mb
green open .kibana_1 VeByZ0ltTsm7evZb_XOEQw 1 0 38 3 950.9kb 950.9kb
green open .monitoring-kibana-6-2019.03.07 Z0xk6Qw9TfmyxIlkCtqKIQ 1 0 988 0 366kb 366kb
green open kibana_sample_data_ecommerce ZMMiH95SQjhd89QJ00iFA 1 0 4675 0 4.8mb 4.8mb
green open .monitoring-kibana-6-2019.03.03 PXlIiS1TQgeYPcLMLMZBrg 1 0 34548 0 5.1mb 5.1mb
green open .monitoring-es-6-2019.03.05 R6Mm64HfRl0cdBZqphdfjQ 1 0 202382 394 103mb 103mb
green open .monitoring-es-6-2019.03.07 f7ephFhcSUuRTZ4hzIFM-Q 1 0 23746 120 13.7mb 13.7mb
```

## 2.7.2 类型/表

1) 创建类型/表，如下在索引 eshop\_product\_ent 中创建类型（表）ent

```
curl -XPUT 10.28.93.179:9200/eshop_product_ent/ent/_mapping \
-H 'Content-Type: application/json' \
-d'
{
  "properties": {
    "subName": {
      "type": "text",
      "analyzer": "ik_max_word",
      "search_analyzer": "ik_smart"
    },
    "name": {
      "type": "text",
      "analyzer": "ik_max_word",
      "search_analyzer": "ik_smart"
    },
    "brandName": {
      "type": "text",
      "analyzer": "ik_max_word",
      "search_analyzer": "ik_smart"
    },
    "secondCategoryName": {
      "type": "text",
      "analyzer": "ik_max_word",
      "search_analyzer": "ik_smart"
    },
    "thirdCategoryName": {
      "type": "text",
      "analyzer": "ik_max_word",
      "search_analyzer": "ik_smart"
    }
  }
}
```

```
}
}
```

2) 查看类型的定义, 类似于 mysql 中查看表结构

```
curl -X GET http://10.28.93.179:9200/quzl/test/_mapping
```

查看类型的定义, 类似于 mysql 中查看表结构

3) 从 6.0 开始单个所以只能存储一个类型/表, 参考官方的说明

<https://www.elastic.co/guide/en/elasticsearch/reference/current/removal-of-types.html>

4) 搜索, 类似 mysql 中的 select

默认显示 10 条

```
curl -XGET '127.0.0.1:9200/eshop_product_ent/ent/_search?pretty'
```

```
curl -XGET '127.0.0.1:9200/eshop_product_ent/ent/_search?size=1&pretty'
```

```
curl -XGET '127.0.0.1:9200/eshop_product_ent/ent/_search?size=1&from=5&pretty'
```

```
curl -XGET '127.0.0.1:9200/eshop_product_ent/ent/_search?pretty&q=price:1000'
```

5) 清空文档/批量删除文档

```
curl -X POST "localhost:9200/.monitoring-es-6-2019.03.01/doc/_delete_by_query?pretty" -H 'Content-Type: application/json' -d'
```

```
{
  "query": {
    "match_all": {}
  }
}
```

```
[root@sdb1 html]# curl -XGET '127.0.0.1:9200/_cat/indices/.monitoring-es-6-2019.03.01?pretty&v'
health status index      uuid      pri rep docs.count docs.deleted store.size pri.store.size
green open   .monitoring-es-6-2019.03.01 dqSfDKF6T32lX_cNmPqDNA 1 0 700972 13540 348.5mb 348.5mb
[root@sdb1 html]#
[root@sdb1 html]#
[root@sdb1 html]# curl -X POST "localhost:9200/.monitoring-es-6-2019.03.01/doc/_delete_by_query?pretty" -H 'Content-Type: application/json' -d'
> {
>   "query": {
>     "match_all": {}
>   }
> }'
>
{
  "took" : 139538,
  "timed_out" : false,
  "total" : 700972,
  "deleted" : 700972,
  "batches" : 701,
  "version_conflicts" : 0,
  "noops" : 0,
  "retries" : {
    "bulk" : 0,
    "search" : 0
  },
  "throttled_millis" : 0,
  "requests_per_second" : -1.0,
  "throttled_until_millis" : 0,
  "failures" : [ ]
}
[root@sdb1 html]#
[root@sdb1 html]# curl -XGET '127.0.0.1:9200/_cat/indices/.monitoring-es-6-2019.03.01?pretty&v'
health status index      uuid      pri rep docs.count docs.deleted store.size pri.store.size
green open   .monitoring-es-6-2019.03.01 dqSfDKF6T32lX_cNmPqDNA 1 0 0 0 348.5mb 348.5mb
[root@sdb1 html]#
[root@sdb1 html]#
```

### 2.7.3 分片和副本

es 默认是 5 个分片(number\_of\_shards, 索引创建后不能修改), 1 个副本(number\_of\_replicas, 可随时修改),

1) 查看分片和副本情况:

```
curl -XGET '10.80.225.121:9200/_cat/shards'
```

```
root@iZuf652coo4lmvft5cjo9nZ:~# curl -XGET '10.28.93.179:9200/_cat/shards'
eshop_product_ent 2 r STARTED 218855 125.4mb 10.28.96.143 node-1
eshop_product_ent 2 p STARTED 218855 126.2mb 10.80.225.121 node-2
eshop_product_ent 1 p STARTED 218989 125.9mb 10.28.93.179 node-3
eshop_product_ent 1 r STARTED 218989 125.7mb 10.80.225.121 node-2
eshop_product_ent 4 r STARTED 218570 125.2mb 10.28.96.143 node-1
eshop_product_ent 4 p STARTED 218570 125.9mb 10.28.93.179 node-3
eshop_product_ent 3 p STARTED 219439 126mb 10.28.96.143 node-1
eshop_product_ent 3 r STARTED 219439 125.5mb 10.28.93.179 node-3
eshop_product_ent 0 p STARTED 218804 125.1mb 10.28.96.143 node-1
eshop_product_ent 0 r STARTED 218804 124.7mb 10.80.225.121 node-2
root@iZuf652coo4lmvft5cjo9nZ:~#
```

p 表示主分片, r 表示副本

2) 创建索引时指定分片数量和副本数量:

```
curl -XPUT 10.28.93.179:9200/quzl/ \
-H 'Content-Type: application/json' \
-d'
{
  "settings": {
    "number_of_shards": 3,
    "number_of_replicas": 0
  }
}'
```

```
root@iZuf652coo4lmvft5cjo9nZ:~# curl -XPUT 10.28.93.179:9200/quzl/ \
> -H 'Content-Type: application/json' \
> -d'
> {
>   "settings": {
>     "number_of_shards": 3,
>     "number_of_replicas": 0
>   }
> }'
{"acknowledged":true,"shards_acknowledged":true,"index":"quzl"}root@iZuf652coo4lmvft5cjo9nZ:~#
root@iZuf652coo4lmvft5cjo9nZ:~#
root@iZuf652coo4lmvft5cjo9nZ:~#
root@iZuf652coo4lmvft5cjo9nZ:~# curl -X GET http://10.28.93.179:9200/_cat/indices
green open quzl kHkS269lT8-43x_NHNaHoQ 3 0 0 0 690b 690b
green open eshop_product_ent GLMLx2WHT7GZrqGctxdHAA 5 1 1094657 0 1.2gb 629.4mb
root@iZuf652coo4lmvft5cjo9nZ:~#
```

3) 修改索引副本数量, 默认副本数量是 1

```
curl -XPUT 10.28.93.179:9200/quzl/_settings \
-H 'Content-Type: application/json' \
-d'
{
  "number_of_replicas": 0
}'
```

```

root@iZuf652coo4lmvft5cjo9nZ:~#
root@iZuf652coo4lmvft5cjo9nZ:~# curl -XPUT 10.28.93.179:9200/_settings -H 'Content-Type: application/json' -d'
{
  "number_of_replicas": 0
}'
{"acknowledged":true}root@iZuf652coo4lmvft5cjo9nZ:~#
root@iZuf652coo4lmvft5cjo9nZ:~#
root@iZuf652coo4lmvft5cjo9nZ:~#
root@iZuf652coo4lmvft5cjo9nZ:~# curl -X GET http://10.28.93.179:9200/_cat/indices
green open quzl          fvAb0_lxQt6XLl1RMqPyBg 5 0      0 0 1.1kb  1.1kb
green open eshop_product_ent GLMLx2WHT7GZrqGctxdHAA 5 1 1094657 0 1.2gb 629.4mb
root@iZuf652coo4lmvft5cjo9nZ:~#
root@iZuf652coo4lmvft5cjo9nZ:~#
root@iZuf652coo4lmvft5cjo9nZ:~#
root@iZuf652coo4lmvft5cjo9nZ:~# curl -XPUT 10.28.93.179:9200/_settings -H 'Content-Type: application/json' -d'
{
  "number_of_replicas": 3
}'
{"acknowledged":true}root@iZuf652coo4lmvft5cjo9nZ:~#
root@iZuf652coo4lmvft5cjo9nZ:~#
root@iZuf652coo4lmvft5cjo9nZ:~# curl -X GET http://10.28.93.179:9200/_cat/indices
yellow open quzl          fvAb0_lxQt6XLl1RMqPyBg 5 3      0 0 3kb  1.2kb
green open eshop_product_ent GLMLx2WHT7GZrqGctxdHAA 5 1 1094657 0 1.2gb 629.4mb
root@iZuf652coo4lmvft5cjo9nZ:~#
root@iZuf652coo4lmvft5cjo9nZ:~#

```

## 2.7.4 cat 查看

查看可以 cat 哪些

curl -XGET '10.28.93.179:9200/\_cat'

```

root@iZuf652coo4lmvft5cjo9nZ:~#
root@iZuf652coo4lmvft5cjo9nZ:~# curl -XGET '10.28.93.179:9200/_cat'
=^.=
/_cat/allocation
/_cat/shards
/_cat/shards/{index}
/_cat/master
/_cat/nodes
/_cat/tasks
/_cat/indices
/_cat/indices/{index}
/_cat/segments
/_cat/segments/{index}
/_cat/count
/_cat/count/{index}
/_cat/recovery
/_cat/recovery/{index}
/_cat/health
/_cat/pending_tasks
/_cat/aliases
/_cat/aliases/{alias}
/_cat/thread_pool
/_cat/thread_pool/{thread_pools}
/_cat/plugins
/_cat/fielddata
/_cat/fielddata/{fields}
/_cat/nodeattrs
/_cat/repositories
/_cat/snapshots/{repository}
/_cat/templates
root@iZuf652coo4lmvft5cjo9nZ:~#

```

查看插件

curl -XGET '10.28.93.179:9200/\_cat/plugins'

```
root@iZuf652c0o4lmvft5cjo9nZ:~# curl -XGET '10.28.93.179:9200/_cat/plugins'
node-2 analysis-ik 6.5.2
node-1 analysis-ik 6.5.2
node-3 analysis-ik 6.5.2
root@iZuf652c0o4lmvft5cjo9nZ:~#
```

查看节点

```
curl -XGET http://10.28.96.143:9200/_cat/nodes?v
```

```
root@iZuf652c0o4lmvft5cjo9nZ:~# curl -XGET http://10.28.96.143:9200/_cat/nodes?v
ip heap.percent ram.percent cpu load_1m load_5m load_15m node.role master name
10.80.225.121 38 64 1 0.00 0.03 0.01 mdi * node-2
10.28.96.143 53 70 1 0.10 0.11 0.04 mdi - node-1
10.28.93.179 7 62 3 0.12 0.13 0.09 mdi - node-3
root@iZuf652c0o4lmvft5cjo9nZ:~#
```

比如使用如下命令，查看集群的健康状态

```
curl -XGET 'http://10.28.96.143:9200/_cat/health?v'
```

```
root@iZuf652c0o4lmvft5cjo9nZ:~# curl -XGET http://10.28.96.143:9200/_cat/health?v
epoch timestamp cluster status node.total node.data shards pri relo init unassign pending_tasks max_task_wait_time active_shards_percent
1551922121 01:28:41 lx2-es-cluster green 3 3 10 5 0 0 0 0 0 100.0%
root@iZuf652c0o4lmvft5cjo9nZ:~#
```

查看文档数量

```
curl -XGET '10.80.225.121:9200/_cat/count'
```

## 2.8 集群部署

我选择三个节点部署集群，配置文件中配置信息如下：/etc/elasticsearch/elasticsearch.yml  
以下 6 个配置项中，node.name 要求各节点必须不同，network.host 可配置成各自的 ip，也可以都写成 0.0.0.0，其他选择都相同。

```
cluster.name: lx2-es-cluster
node.name: node-2
path.data: /data/elasticsearch
path.logs: /var/log/elasticsearch
network.host: 10.80.225.121
discovery.zen.ping.unicast.hosts: ["10.28.96.143", "10.80.225.121", "10.28.93.179"]
```

配置后使用如下命令简单查看集群状态，\*号表示是主节点

```
curl -XGET 'http://10.80.225.121:9200/_cat/nodes?pretty'
```

```
root@iZuf652c0o4lmvft5cjo9nZ:/etc/elasticsearch#
root@iZuf652c0o4lmvft5cjo9nZ:/etc/elasticsearch# curl -XGET 'http://10.80.225.121:9200/_cat/nodes?pretty'
10.28.96.143 7 38 1 0.03 0.08 0.03 mdi * node-1
10.80.225.121 7 48 2 0.17 0.15 0.09 mdi - node-2
10.28.93.179 11 45 2 0.60 0.35 0.18 mdi - node-3
root@iZuf652c0o4lmvft5cjo9nZ:/etc/elasticsearch#
```

端口监听如下，集群内部使用端口 9300 通信，9200 端口提供服务。

```
root@iZuf652c0o4lmvft5cjo9nZ:/etc/elasticsearch#
root@iZuf652c0o4lmvft5cjo9nZ:/etc/elasticsearch# netstat -tnlp|grep 16789
tcp 0 0 10.80.225.121:9200 0.0.0.0:* LISTEN 16789/java
tcp 0 0 10.80.225.121:9300 0.0.0.0:* LISTEN 16789/java
root@iZuf652c0o4lmvft5cjo9nZ:/etc/elasticsearch#
```

```

root@iZuf652c004lmvft5cjo9nZ:/etc/elasticsearc# curl -XGET 'http://10.28.96.143:9200/_cat/health?pretty'
1551320209 02:16:49 lx2-es-cluster green 3 3 0 0 0 0 0 - 100.0%
root@iZuf652c004lmvft5cjo9nZ:/etc/elasticsearch#
root@iZuf652c004lmvft5cjo9nZ:/etc/elasticsearch#
root@iZuf652c004lmvft5cjo9nZ:/etc/elasticsearch# curl -XGET 'http://10.28.96.143:9200/_cat?pretty'
=^_^=
/_cat/allocation
/_cat/shards
/_cat/shards/{index}
/_cat/master
/_cat/nodes
/_cat/tasks
/_cat/indices
/_cat/indices/{index}
/_cat/segments
/_cat/segments/{index}
/_cat/count
/_cat/count/{index}
/_cat/recovery
/_cat/recovery/{index}
/_cat/health
/_cat/pending_tasks
/_cat/aliases
/_cat/aliases/{alias}
/_cat/thread_pool
/_cat/thread_pool/{thread_pools}
/_cat/plugins
/_cat/fielddata
/_cat/fielddata/{fields}
/_cat/nodeattrs
/_cat/repositories
/_cat/snapshots/{repository}
/_cat/templates
root@iZuf652c004lmvft5cjo9nZ:/etc/elasticsearch#

```

## 第三章 Logstash 部署

官方文档:

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

### 3.1 下载

下载地址

<https://www.elastic.co/downloads/logstash>

这里使用 rpm 安装

wget <https://artifacts.elastic.co/downloads/logstash/logstash-6.5.3.rpm>

如果是 ubuntu 系统, 使用 deb 包安装。

### 3.2 安装

安装、运行、添加到开机启动项。

```
rpm -ivh logstash-6.5.3.rpm
```

或者

```
rpm -ivh https://artifacts.elastic.co/downloads/logstash/logstash-6.5.3.rpm
```

该文件 150M 左右

```
systemctl daemon-reload
systemctl enable logstash
systemctl start logstash
systemctl status logstash
```

### 3.3 配置文件

#### 3.3.1 java 变量-非必需

如果使用 oracle-jdk, 需要添加如下一行配置

```
JAVA_HOME=/usr/local/java
```

到文件 /usr/share/logstash/bin/logstash.lib.sh 中。

如果使用 yum/apt 等安装的 openjdk, 则不需要这一步。



### 3.3.2 配置 es 的位置信息

### 3.3.3 hello world 测试

输入以下命令

```
./logstash -e 'input{stdin{}} output{stdout{codec=>rubydebug}}'
```

或者

```
./logstash -e "
```

输入 hello world

返回 rubydebug 格式的内容，如下图：

```
[root@tidbl bin]# ./logstash -e 'input{stdin{}} output{stdout{codec=>rubydebug}}'
hello world
WARNING: Could not find logstash.yml which is typically located in $LS_HOME/config or /etc/logstash. You can
Could not find log4j2 configuration at path /usr/share/logstash/config/log4j2.properties. Using default confi
[WARN ] 2018-12-18 16:59:42.591 [LogStash::Runner] multilocal - Ignoring the 'pipelines.yml' file because mod
[INFO ] 2018-12-18 16:59:42.617 [LogStash::Runner] runner - Starting Logstash {"logstash.version"=>"6.5.3"}
[INFO ] 2018-12-18 16:59:42.658 [LogStash::Runner] agent - No persistent UUID file found. Generating new UUID
uuid"}
[INFO ] 2018-12-18 16:59:47.661 [Converge PipelineAction::Create<main>] pipeline - Starting pipeline {:pipeli
delay"=>50}
[INFO ] 2018-12-18 16:59:47.851 [Converge PipelineAction::Create<main>] pipeline - Pipeline started successfu
The stdin plugin is now waiting for input:
[INFO ] 2018-12-18 16:59:47.944 [Ruby-0-Thread-1: /usr/share/logstash/lib/bootstrap/environment.rb:6] agent -
=>[]
[INFO ] 2018-12-18 16:59:48.725 [Api Webserver] agent - Successfully started Logstash API endpoint {:port=>96
{
  "message" => "hello world",
  "@timestamp" => 2018-12-18T08:59:47.990Z,
  "host" => "tidbl",
  "@version" => "1"
}
```

## 3.4 插件管理

### 3.4.1 插件操作

列出已经安装的插件：

```
./logstash-plugin list
```

安装插件：

```
./logstash-plugin install **
```

更新来源：

<https://github.com/logstash-plugins?page=1>

升级插件：

```
./logstash-plugin update **
```

例如安装 jdbc 插件：

```
./bin/logstash-plugin install logstash-input-jdbc  
[root@iZuf6h75wm95cngv3glncvZ logstash]# ./bin/logstash-plugin install logstash-input-jdbc  
Validating logstash-input-jdbc  
Installing logstash-input-jdbc  
Installation successful  
[root@iZuf6h75wm95cngv3glncvZ logstash]#  
[root@iZuf6h75wm95cngv3glncvZ logstash]# ./bin/logstash-plugin list|grep jdbc  
logstash-filter-jdbc_static  
logstash-filter-jdbc_streaming  
logstash-input-jdbc  
[root@iZuf6h75wm95cngv3glncvZ logstash]#
```

### 3.4.2 geoip 插件

geoip 包含如下字段可以用于显示：

```
"geoip" => {  
    "ip" => "112.90.16.4",  
    "country_code2" => "CN",  
    "country_code3" => "CHN",  
    "country_name" => "China",  
    "continent_code" => "AS",  
    "region_name" => "30",  
    "city_name" => "Guangzhou",  
    "latitude" => 23.11670000000001,  
    "longitude" => 113.25,  
    "timezone" => "Asia/Chongqing",  
    "region_name " => "Guangdong",  
    "location" => [  
    [0] 113.25,  
    [1] 23.116700000000001  
    ]  
}
```

以下官方文档中说

<https://www.elastic.co/guide/en/logstash/current/plugins-filters-geoip.html>

有以下字段

ip, IP 地址

city\_name, 城市名, 比如 Guangzhou

continent\_code, 洲际简称, 两个字母, 比如 AS (亚洲)

country\_code2, 国家/地区简称, 两个字母, 比如 CN (中国)

country\_code3, 国家/地区简称, 三个字母, 比如 CHN (中国)

country\_name, 国家/地区全程, 比如 China(中国)

dma\_code, 市场区号, 只支持美国和加拿大

latitude, 有符号的双精度纬度

longitude, 有符号的双精度经度

postal\_code, 邮编, FSA 或者 Zip 编码

region\_name, 省份名称, 比如 Guangdong(广东)  
timezone, 时区, 比如 Asia/Shanghai (亚洲上海)

如果不需要的话, 选择不显示:

### 3.4.3 useragent 插件

该插件包含以下信息:

```
"ua" => {  
  "patch" => "2883",  
  "os" => "Windows 7",  
  "major" => "55",  
  "minor" => "0",  
  "name" => "Chrome",  
  "os_name" => "Windows 7",  
  "device" => "Other"  
}
```

## 3.5 nginx 日志上报

### 3.5.1 日志格式

两种方法:

#### 1) 一般日志

logstash 处理时使用 grok 表达式进行匹配

nginx 日志 grok 正则匹配表达式含义:

<https://www.cnblogs.com/stozen/p/5638369.html>

#### 2) json 日志

nginx 输出 json 格式的日志, 可直接上报, 比较简单。

我这里使用 json 的方式定义如下:

```
log_format logstash_json '{  
    "@timestamp": "$time_iso8601", '  
    "server_name": "$server_name", '  
    "remote_addr": "$remote_addr", '  
    "remote_user": "$remote_user", '  
    "body_bytes_sent": "$body_bytes_sent", '  
    "cookie_JSESSIONID": "$cookie_JSESSIONID", '  
    "status": "$status", '  
    "request": "$request", '  
    "request_method": "$request_method", '
```

```

        "http_referrer": "$http_referer", '
        "body_bytes_sent": $body_bytes_sent, '
        "http_x_forwarded_for": "$http_x_forwarded_for", '
        "http_user_agent": "$http_user_agent", '
        "upstream_response_time": "$upstream_response_time", '
        "request_time": $request_time'
    '};

```

logstash 配置文件如下:

```

input {
    file {
        path => "/usr/local/nginx/logs/access.json.log"
        type => "nginx-ua-access"
        codec => "json"
        start_position => "beginning"
    }
}
filter {
    geoip {
        source => "remote_addr"
        target => "geoip"
        #database => "/etc/logstash/GeoLite2-City.mmdb"
        remove_field => ["[geoip][postal_code]", "[geoip][dma_code]", "[geoip][country_code2]", "[geoip][country_code3]", "[geoip][longitude]", "[geoip][latitude]", "[geoip][region_code]", "[geoip][timezone]"]
        #add_field => ["[geoip][coordinates]", "%{[geoip][longitude]}"]
        #add_field => ["[geoip][coordinates]", "%{[geoip][latitude]}"]
    }
    useragent {
        source => "http_user_agent"
        target => "agent"
        remove_field => ["[agent][build]", "[agent][os_name]", "[agent][device]", "[agent][minor]", "[agent][patch]", "[agent][os_minor]", "[agent][major]", "[agent][os_major]"]
    }

    mutate {
        #convert => ["[geoip][coordinates]", "float"]
        #convert => ["[geoip][longitude]", "float"]
        #convert => ["[geoip][latitude]", "float"]
        convert => ["upstream_response_time", "float"]
        convert => ["request_time", "float"]
        #convert => ["body_bytes_sent", "integer"]
        remove_field => ["http_user_agent"]
    }
}

```

```

}
output {
  elasticsearch {
    hosts => ["172.16.6.11:9200"]
    index => "logstash-nginx-%{+YYYY.MM.dd}"
  }
}

```

### 3.5.2 绘制坐标地图

- 1) logstash 上报时，索引必须使用 logstash 开头，比如

```

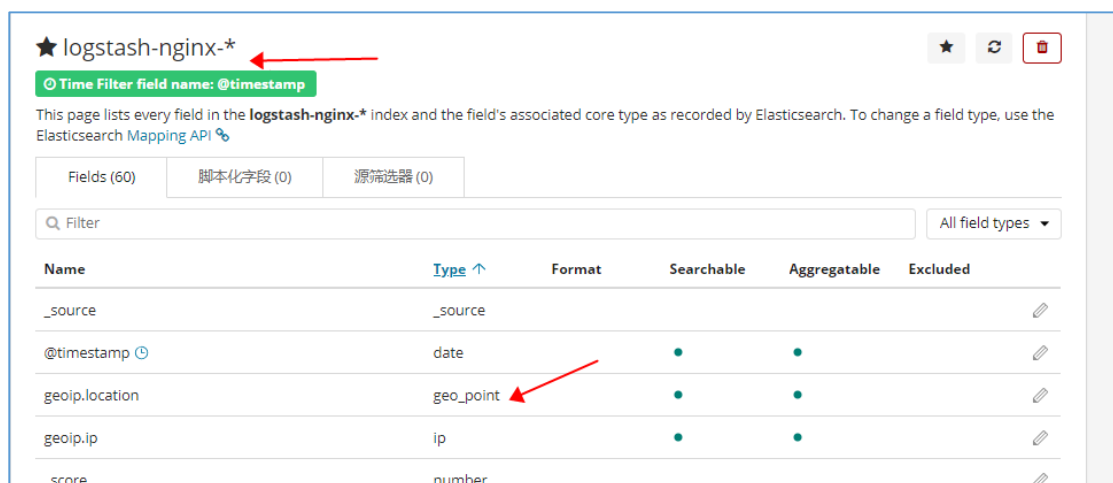
output {
  elasticsearch {
    hosts => ["172.16.6.11:9200"]
    index => "logstash-nginx-%{+YYYY.MM.dd}"
  }
}

```

因为 es 内置模板中，要求索引名必须是 logstash-开头的，可通过以下连接进行确认

[http://172.16.6.11:9200/\\_template?pretty](http://172.16.6.11:9200/_template?pretty)

- 2) 确保 geoip.location 的类型是 geo\_point，如下图



★ logstash-nginx-\*

Time Filter field name: @timestamp

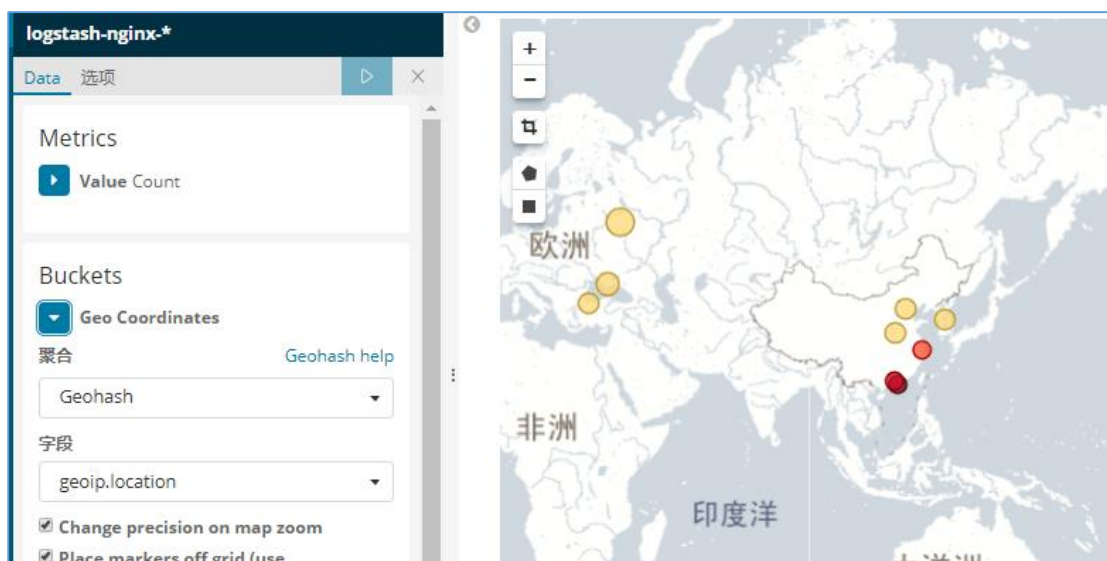
This page lists every field in the **logstash-nginx-\*** index and the field's associated core type as recorded by Elasticsearch. To change a field type, use the [Elasticsearch Mapping API](#)

Fields (60)    脚本化字段 (0)    源筛选器 (0)

Filter    All field types

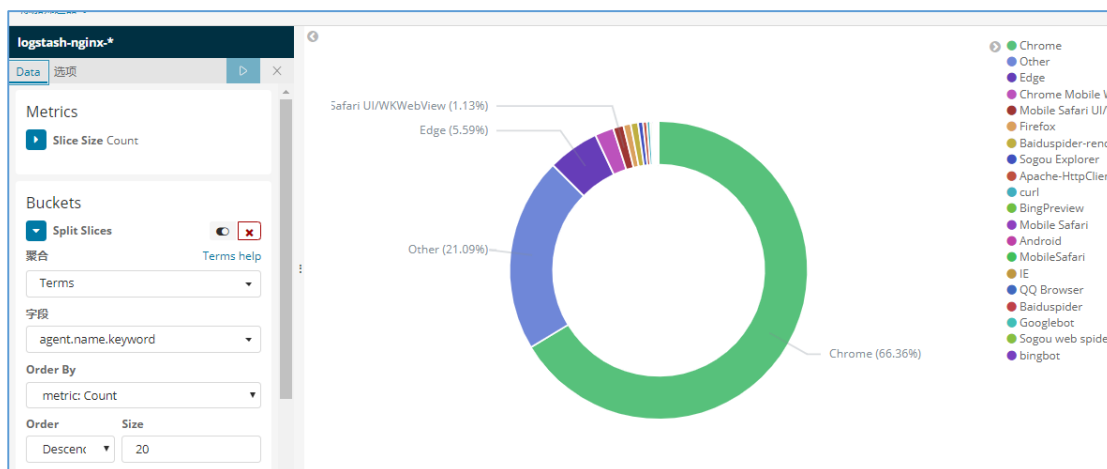
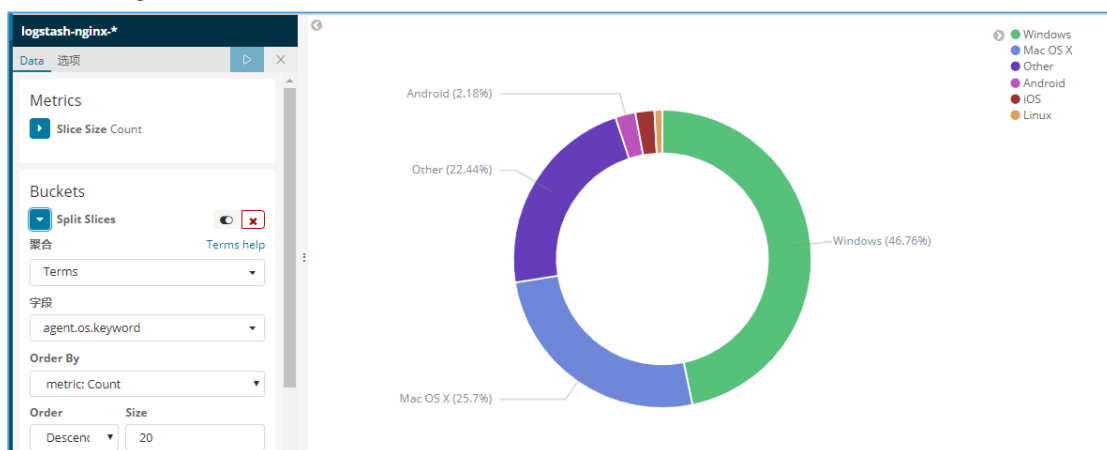
Name	Type ↑	Format	Searchable	Aggregatable	Excluded
_source	_source				
@timestamp	date		•	•	
geoip.location	geo_point		•	•	
geoip.ip	ip		•	•	
_score	number				

- 3) 默认使用官方的地图，可修改为高德地图，中文的方法是在 kibana 的配置文件/etc/kibana/kibana.yml 最后添加如下五行：  
 tilemap.url:  
 'http://webrd02.is.autonavi.com/appmaptile?lang=zh\_cn&size=1&scale=1&style=7&x={x}&y={y}&z={z}'  
 添加后，重新加载 kibana 服务，重新绘制地图，对应已经绘制完成的地图是不生效的。
- 4) 绘图如下：



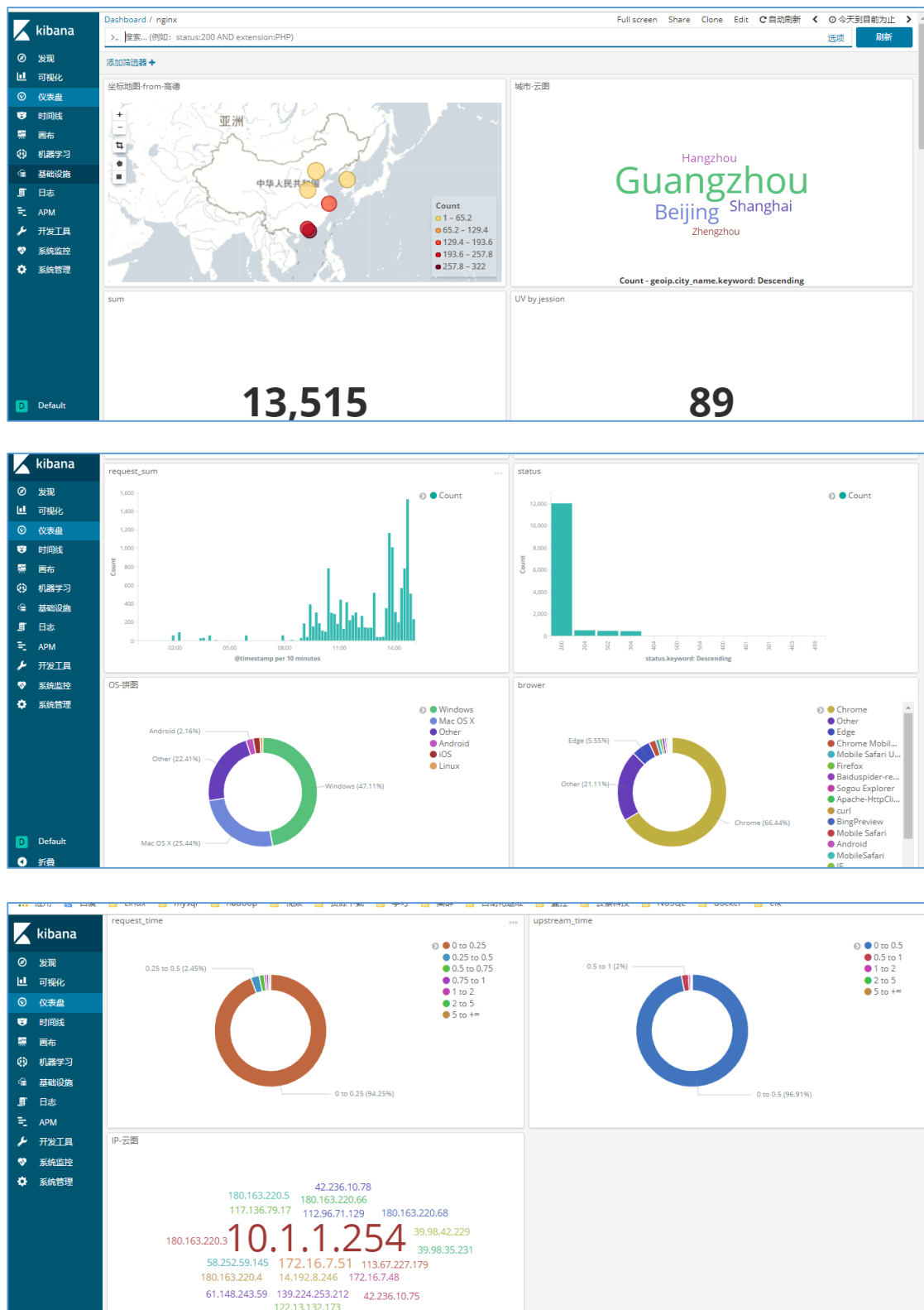
### 3.5.3 绘制客户端信息图

使用 useragent 插件捕捉到的相关信息进行绘制。如下图



### 3.5.4 自定义仪表盘

很灵活，可根据需要进行随意组合，比如下面是我测试用的



### 3.6 spring boot 日志上报

方法一：通过 logstash 使用 file 模块上报，代码无需变动

方法二：spring boot 打包出 json 格式的日志，再使用 logstash 上报，需要变动代码

方法三：spring boot 直接通过 tcp 协议，上报给 logstash，程序本地无需打印日志



## 第四章 Kibana 部署

官方文档:

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

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

### 4.1 下载

下载地址

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

这里使用 rpm 安装

wget [https://artifacts.elastic.co/downloads/kibana/kibana-6.5.3-x86\\_64.rpm](https://artifacts.elastic.co/downloads/kibana/kibana-6.5.3-x86_64.rpm)

如果是 ubuntu 系统，使用 deb 包安装。

### 4.2 安装

安装、运行、添加到开机启动项。

```
rpm -ivh kibana-6.5.3-x86_64.rpm
```

或者

```
rpm -ivh https://artifacts.elastic.co/downloads/kibana/kibana-6.5.3-x86\_64.rpm
```

该文件 200M 左右

```
systemctl daemon-reload
systemctl enable kibana
systemctl start kibana
systemctl status kibana
```

### 4.3 配置文件

#### 4.3.1 监听的 IP 和端口

kibana 默认监听 127.0.0.1:5601，可根据需要进行修改/etc/kibana/kibana.yml 中的以下两行配置：

```
#server.port: 5601
#server.host: "localhost"
```

例如修改为：

```
server.port: 5600
server.host: "0.0.0.0"
```

### 4.3.2 汉化

```
wget https://github.com/anbai-inc/Kibana_Hanization/archive/master.zip
unzip master.zip
cd Kibana_Hanization-master
python main.py /usr/share/kibana
```

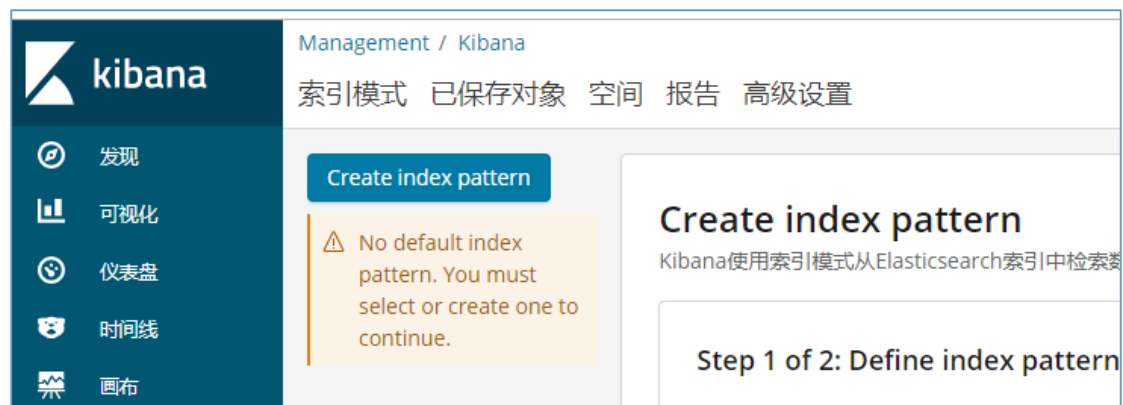
```
[root@tidbl Kibana_Hanization-master]# python main.py /usr/share/kibana/
文件[/usr/share/kibana/node_modules/@elastic/eui/dist/eui.js]已翻译。
文件[/usr/share/kibana/node_modules/@elastic/eui/dist/eui.min.js]已翻译。
文件[/usr/share/kibana/node_modules/@elastic/eui/es/components/search_bar/search_box.js]已翻译。
文件[/usr/share/kibana/node_modules/@elastic/eui/lib/components/search_bar/search_box.js]已翻译。
文件[/usr/share/kibana/node_modules/@elastic/eui/src/components/search_bar/search_box.js]已翻译。

文件[/usr/share/kibana/node_modules/x-pack/plugins/apm/public/components/app/TransactionOverview/DynamicBa
文件[/usr/share/kibana/node_modules/x-pack/plugins/canvas/index.js]已翻译。
文件[/usr/share/kibana/node_modules/x-pack/plugins/canvas/canvas_plugin/renderers/all.js]已翻译。
文件[/usr/share/kibana/node_modules/x-pack/plugins/canvas/canvas_plugin/uis/arguments/all.js]已翻译。
文件[/usr/share/kibana/node_modules/x-pack/plugins/canvas/canvas_plugin/uis/datasources/all.js]已翻译。
文件[/usr/share/kibana/node_modules/x-pack/plugins/canvas/public/register_feature.js]已翻译。
文件[/usr/share/kibana/node_modules/x-pack/plugins/infra/index.js]已翻译。
```

汉化完成后，重启 kibana

```
systemctl restart kibana
```

使用浏览器登陆查看，汉化成功。



### 4.3.3 KILL\_ON\_STOP\_TIMEOUT

修改以下配置文件

/etc/default/kibana

默认是：

```
KILL_ON_STOP_TIMEOUT=0
```

设置为 0，如果停止失败（比如 es 停止后，kibana 将无法停止，kill -9 也不行），将自动重启。

修改为：

```
KILL_ON_STOP_TIMEOUT=1
```

设置为 1，如果停止失败，将发送 SIGKILL 信号将其停止。

## 4.4 使用

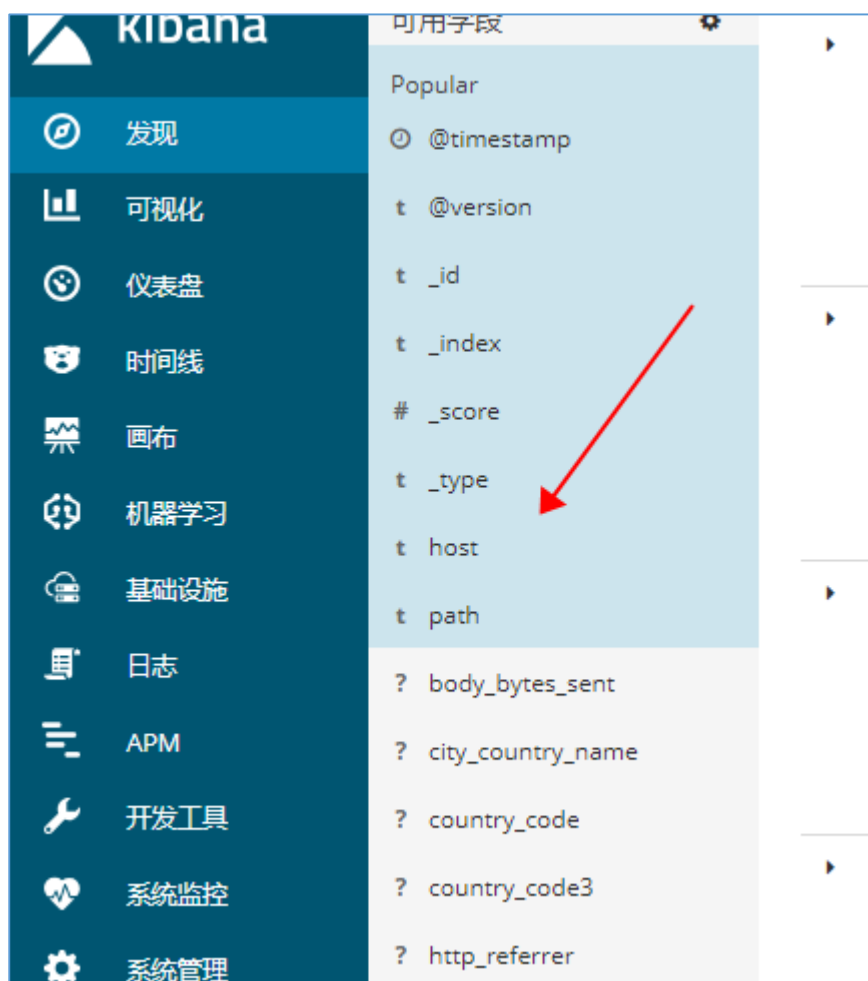
### 4.4.1 查看状态

使用浏览器访问如下 url:

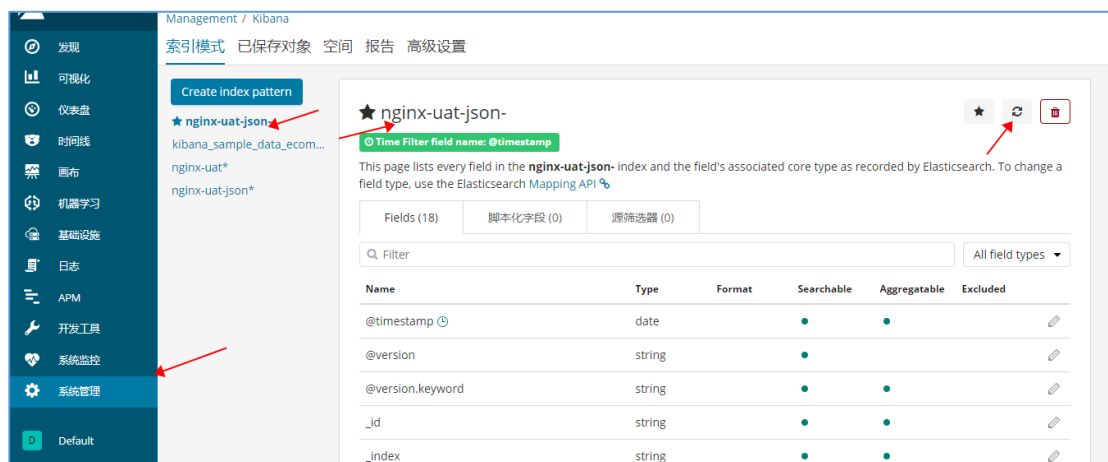
<http://172.16.6.11:5601/status>

### 4.4.2 刷新字段

当一种日志上报后，如下图可以看到多了很多字段



但在可视化绘图等界面中，看不到这些新增的字段，需要刷新下即可显示了，如下图



## 4.5 坐标地图

参考 3.5.2

# 第五章 es 搜索引擎部署

## 5.1 部署 es 集群

参考 2.8 节

## 5.2 部署 logstash 节点

用于将 mysql 数据同步到 es，即关系型数据库内数据传输的 es，参考第三章。

## 5.3 配置

### 5.3.1 es 安装中文分词

参考 2.6 节，注意集群中所有 es 都要安装，安装完成后重启

### 5.3.2 logstash 安装 jdbc 插件

参考 3.4.1

## 5.4 创建索引和 mapping

```
curl -XPUT 10.28.93.179:9200/eshop_product_ent

curl -XPUT 10.28.93.179:9200/eshop_product_ent/ent/_mapping \
-H 'Content-Type: application/json' \
-d'
{
  "properties": {
    "subName": {
      "type": "text",
      "analyzer": "ik_max_word",
      "search_analyzer": "ik_smart"
    },
    "name": {
      "type": "text",
      "analyzer": "ik_max_word",
      "search_analyzer": "ik_smart"
    },
    "brandName": {
      "type": "text",
      "analyzer": "ik_max_word",
      "search_analyzer": "ik_smart"
    },
    "secondCategoryName": {
      "type": "text",
      "analyzer": "ik_max_word",
      "search_analyzer": "ik_smart"
    },
    "thirdCategoryName": {
      "type": "text",
      "analyzer": "ik_max_word",
      "search_analyzer": "ik_smart"
    }
  }
}
```

## 5.1 同步数据

在 logstash 节点， /usr/share/logstash/bin/jdbc 目录下提供如下三个文件：

文件 1：mysql-connector-java-5.1.45.jar

文件 2: jdbc.conf

```
cat jdbc/jdbc.conf
input {
  stdin {}
  jdbc {
    clean_run => true
    jdbc_driver_library => "/usr/share/logstash/bin/jdbc/mysql-connector-java-5.1.45.jar"
    jdbc_driver_class => "com.mysql.jdbc.Driver"
    jdbc_connection_string =>
"jdbc:mysql://10.29.47.173:3306/louxe_eshop_db?useUnicode=true&characterEncoding=utf
-8&zeroDateTimeBehavior=convertToNull&useSSL=false"
    jdbc_user => "entportal"
    jdbc_password => "Cl0udSplus#2018"
    lowercase_column_names => false
    jdbc_paging_enabled => true
    jdbc_page_size => "5000"
    statement_filepath => "/usr/share/logstash/bin/jdbc/product.sql"
  }
}

filter {
  json {
    source => "message"
    remove_field => ["message"]
  }
}

output {
  elasticsearch {
    index => "eshop_product_ent"
    document_type => "ent"
    hosts => "http://10.28.96.143:9200/"
    document_id => "%{indexId}"
  }
  stdout {
    codec => json_lines
  }
}
```

文件 3: product.sql

```
cat jdbc/product.sql
SELECT
  B.ID AS indexId,
  P.ID AS id,
  P.CODE AS code,
```

```

P.NAME AS name,
P.SUB_NAME AS subName,
P.IMG AS img,
P.FIRST_CATEGORY_CODE AS firstCategoryCode,
p.FIRST_CATEGORY_NAME AS firstCategoryName,
p.SECOND_CATEGORY_CODE AS secondCategoryCode,
p.SECOND_CATEGORY_NAME AS secondCategoryName,
p.THIRD_CATEGORY_CODE AS thirdCategoryCode,
p.THIRD_CATEGORY_NAME AS thirdCategoryName,
P.VIEW_COUNT AS viewCount,
P.SALE_COUNT AS saleCount,
P.SUPPLIER_CODE AS supplierCode,
P.SUPPLIER_NAME AS supplierName,
P.SUPPLIER_SKU_ID AS supplierSkuld,
P.IS_SEVEN_RETURN AS sevenReturn,
P.IS_DEPOSIT_PRODUCT AS depositProduct,
P.SELL_TYPE AS sellType,
P.BRAND_CODE AS brandCode,
P.BRAND_NAME AS brandName,
P.SALE_UNIT AS saleUnit,
P.POINT_PROFIT AS pointProfit,
P.SPECIFICATION_NAME AS specificationName,
P.PRODUCT_INDEX AS productIndex,
B.CITY_CODE AS cityCode,
B.PRICE AS price,
B.CREATE_DATE AS createDate,
B.IS_TOP AS top ,
B.SHIPPING_COUNT AS shippingCount,
B.FREIGHT_TEMPLATE_CODE AS freightTemplateCode,
B.LINE_PRICE AS linePrice,
B.FAVOURABLE_DATA AS favourableData,
B.`PLATFORM` AS platform

FROM
  T_ES_PRODUCT P
  LEFT JOIN T_ES_PRODUCT_BUILDING B
    ON P.CODE = B.PRODUCT_CODE
WHERE  b.PLATFORM = 'ENT'
ORDER BY
  ID DESC

```

执行命令:

```
cd /usr/share/logstash/bin
```

```
nohup ./logstash -f jdbc/jdbc.conf --config.reload.automatic > a.txt &
```

观察同步过程:

```
tail -f a.txt
```

或者

```
curl -XGET '10.80.225.121:9200/_cat/indices/eshop_product_ent?v'
```

```
root@iZuf652coo41mvft5cjo9mZ:~# curl -XGET '10.80.225.121:9200/_cat/indices/eshop_product_ent?v'
health status index      uuid                                pri rep docs.count docs.deleted store.size pri.store.size
green open   eshop_product_ent GLMLx2WHT7GZrqGctxdHAA 5 1 1094657 0 1.2gb 629.4mb
root@iZuf652coo41mvft5cjo9mZ:~#
```



参考文档:

权威指南:

<https://es.xiaoleilu.com/index.html>

<http://wiki.jikexueyuan.com/project/elasticsearch-definitive-guide-cn/>