ELK 文档整理

叶剑科(yeke)

一．背景与简介

日志主要包括系统日志、应用程序日志和安全日志。系统运维和开发人员可以通过日志了解服务器软硬件信息、检查配置过程中的错误及错误发生的原因。经常分析日志可以了解服务器的负荷，性能安全性，从而及时采取措施纠正错误。通常，日志被分散的储存不同的设备上。如果你管理数十上百台服务器，你还在使用依次登录每台机器的传统方法查阅日志。这样是不是感觉很繁琐和效率低下。当务之急我们使用集中化的日志管理，例如：开源的 syslog，将所有服务器上的日志收集汇总。集中化管理日志后，日志的统计和检索又成为一件比较麻烦的事情，一般我们使用 grep、awk 和 wc 等 Linux 命令能实现检索和统计，但是对于要求更高的查询、排序和统计等要求和庞大的机器数量依然使用这样的方法难免有点力不从心。

开源实时日志分析 ELK 平台能够完美的解决我们上述的问题，ELK 由 ElasticSearch、Logstash 和 Kiabana 三个开源工具组成。

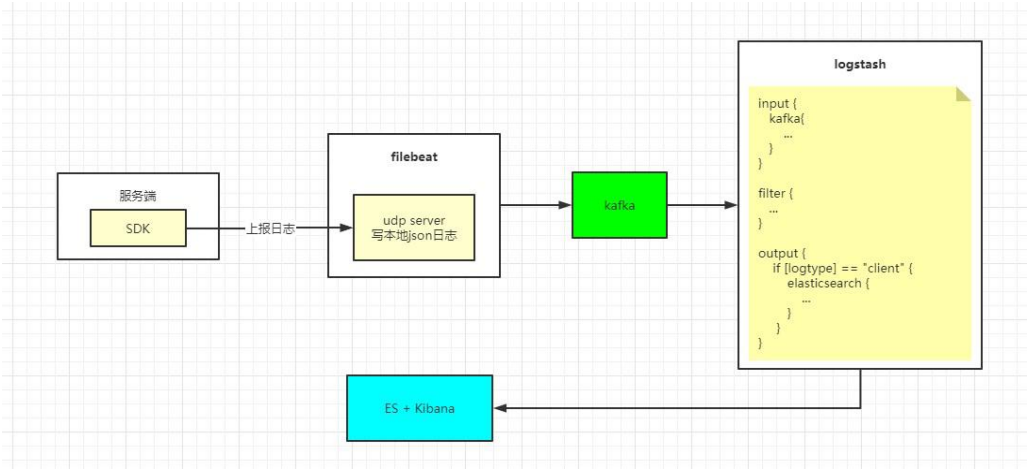
Elasticsearch 是个开源分布式搜索引擎，它的特点有：分布式，零配置，自动发现，索引自动分片，索引副本机制，restful 风格接口，多数据源，自动搜索负载等。

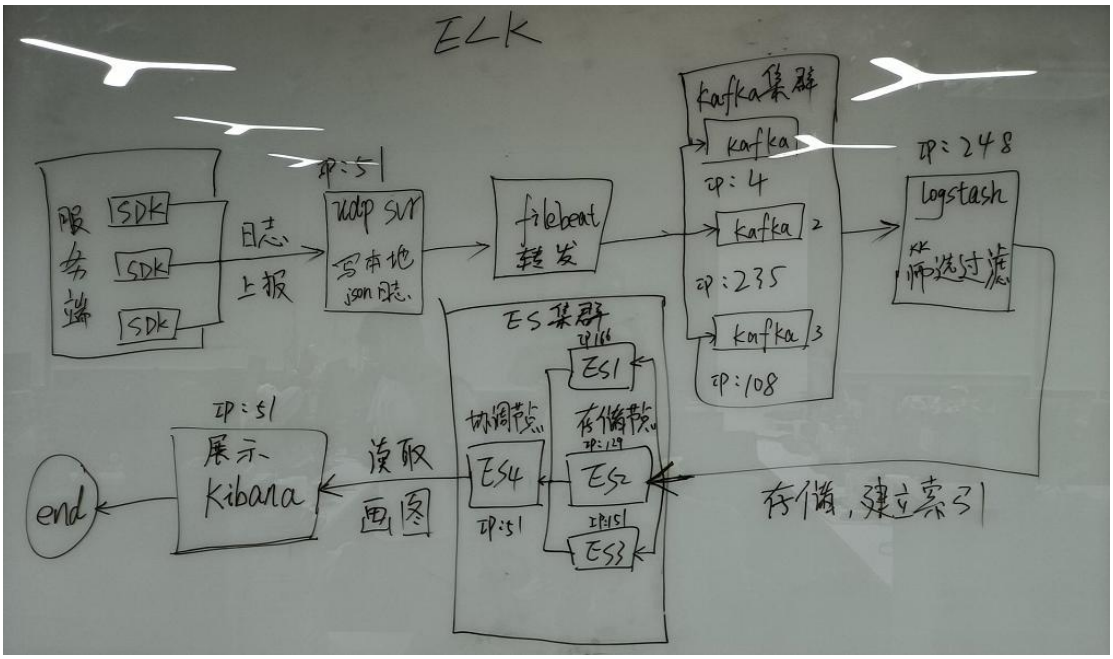
Logstash 是一个完全开源的工具，他可以对你的日志进行收集、过滤，并将其存储供以后使用（如，搜索）。

Kibana 也是一个开源和免费的工具，它 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。

选取的方案：Sdk+udpser+filebeat+kafka+logstash+elasticsearch+kibana 模式

流程图如下：





二．ELK 的安装

2.1 java 安装

elk 依赖于 java(1.8.0 以上)，先安装 java（最新）

查看 java 版本： java -version

安装 java 版本： yum install java

2.2 elk 安装包的下载

下载对应的安装包：https://www.elastic.co/cn/downloads/

注意：不同的平台下载对应的安装包

2.3 elsastic 的安装

mkdir elk

cd ./elk

把 2.2 下载安装包拷到此目录下

解压：

tar -zxvf elasticsearch-7.9.0-linux-x86_64.tar.gz

进入安装目录

cd elasticsearch-7.9.0/bin

开启服务

./elasticsearch

（注意要使用非 root 用户开启 elasticsearch 服务）

后台自启动

./elasticsearch -d

新开窗口查看是否开启

curl 'http://localhost:9200/?pretty'

2.4 logstash 的安装

mkdir elk

cd ./elk

把 2.2 下载安装包拷到此目录下

解压：

tar -zxvf logstash-7.9.0.tar.gz

使用非 root 用户启动

cd ./logstash-7.9.0/bin/

./logstash -b 1500 -w 32 -f ./logstash-7.9.0/config/xxx.conf

2.5 kibana 的安装

mkdir elk

cd /usr/local/elk

把 2.2 下载安装包拷到此目录下

解压：

tar -zxvf kibana-7.9.0-linux-x86_64.tar.gz

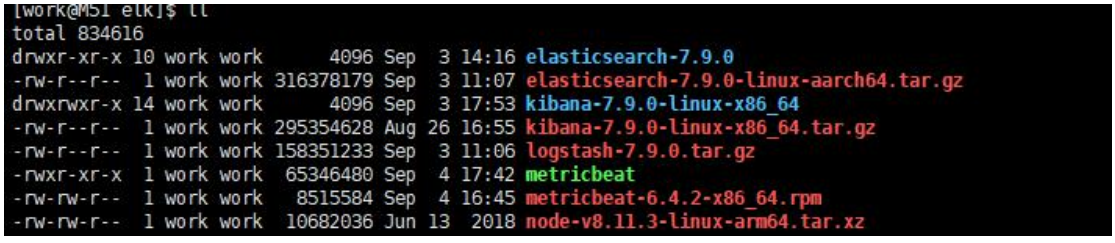
使用非 root 用户启动

./kibana-7.9.0/bin/kibana

2.5 华为鲲鹏云安装

参考文献 2，下载对应最新的安装包

安装包如下；



三．ELK 的配置

3.1 logstash 的配置

在 logstash 的安装目录新建 logstasg.conf

```
input {
  kafka {
    bootstrap_servers                                                    =>
["192.168.1.235:9092,192.168.1.4:9092,192.168.1.108:9092"]
    consumer_threads => 8
    decorate_events => true
    topics => ["zhoujiangtest"]
    codec => "json"
    group_id => "logstash"
    client_id => "logstash3"
    }
}
filter {
  json{
    source => "message"
  }
 mutate {
    remove_field => ["message"]
  }
 mutate { lowercase => [ "server" , "sdk_type" , "event" ] }
}

output {
  elasticsearch {
```

```
    hosts                                                    =>
["192.168.1.166:9200","192.168.1.151:9200","192.168.1.129:9200"]
    index => "logstash-%{sdk_type}-%{server}-%{event}-%{+YYYY.MM.dd}"
    #user => "elastic"
    #password => "vcds8zeiKPoqr0eFzUqr"
    }
}
```

3.2 Es7.9.0 的配置

3.2.1 es 的机器参数配置

第一个问题：
解决：切换到 root 用户，（su root 切换用户）编辑 limits.conf 添加类似如下内容
vi /etc/security/limits.conf
添加如下内容:
* soft nofile 65536
* hard nofile 131072
* soft nproc 2048
* hard nproc 4096

第二个问题：
这个问题大概是说给这个用户分的 1024 不够大，至少需要 4096
解决：切换到 root 用户，进入 limits.d 目录下修改配置文件。
vi /etc/security/limits.d/90-nproc.conf
修改如下内容：
* soft nproc 1024
#修改为
* soft nproc 2048
(这里顺便记下 vi 的编辑和保存，当时忘了，输入命令后进入 vim 编辑器，按 i 进入编辑状态,编辑后 ESC->:->wq(保存退出)或者 q!(直接退出不保存)->回车)

第三个问题：
解决：切换到 root 用户修改配置 sysctl.conf
vi /etc/sysctl.conf
添加下面配置：
vm.max_map_count=655360

并执行命令：
sysctl -p

3.2.2 es 的系统参数配置

节点说明

类型	配置	作用
主节点	node.master: true node.data: false node.ingest: false	负责集群级别的维护操作，集群状态、创建和删除索引、节点上分片的分配操作等等。
数据节点	node.master: false node.data: true node.ingest: false	保存数据的节点。处理和数据相关的请求操作，承载写入和查询的重要节点
Ingest 节点	node.master: false node.data: false node.ingest: true	专注于数据的预处理
协调节点	node.master: false node.data: false node.ingest: false	处理路由请求；处理搜索聚合节点；分发批里索引请求。

打开 config/elasticsearch.yml 配置如下：

#集群名

cluster.name: es-yb-system

#节点名

node.name: node-51

node.master: false

node.data: false

node.ingest: false

search.remote.connect: false

network.host: 192.168.1.51

http.host: 192.168.1.51

http.port: 9200

#log 目录

path.data: /home/work/elk/elasticsearch-7.9.0/data

path.logs: /home/work/elk/elasticsearch-7.9.0/logs

```
bootstrap.memory_lock: false
bootstrap.system_call_filter: false

discovery.seed_hosts:
["192.168.1.166","192.168.1.151","192.168.1.129","192.168.1.250","192.
168.1.167","192.168.1.110","192.168.1.51"]
## es7.x 之后新增的配置，初始化一个新的集群时需要此配置来选举 master
#cluster.initial_master_nodes:
["192.168.1.250","192.168.1.167","192.168.1.110"]
cluster.initial_master_nodes: ["192.168.1.110"]
```

3.3 Kibana 密码配置

打开 vim ./config/kibana.yml 配置如下：

```
server.port: 5601
server.host: "192.168.1.51"
server.name: "kibana"
# 这里可以配置多个
elasticsearch.hosts: ["http://192.168.1.51:9200"]
kibana.index: ".kibana"
kibana.defaultAppId: "home"
# #中文显示
i18n.locale: "zh-CN"

#配置账户密码
#elasticsearch.username: "elastic"
#elasticsearch.password: "vcds8zeiKPoqr0eFzUqr"
```

四．Elk 的查询与测试

4.1 es 的命令行操作

4.1.1 查看集群健康信息

查看 es 集群状态

curl 'http://192.168.1.51:9200/_cat/health?v'

curl 'http://192.168.1.51:9200/_cluster/health?pretty'

关键指标说明：

status：集群状态，分为 green、yellow 和 red。

number_of_nodes/number_of_data_nodes:集群的节点数和数据节点数。

active_primary_shards：集群中所有活跃的主分片数。

active_shards：集群中所有活跃的分片数。

relocating_shards：当前节点迁往其他节点的分片数量，通常为 0，当有节点加入或者退出时该值会增加。

initializing_shards：正在初始化的分片。

unassigned_shards：未分配的分片数，通常为 0，当有某个节点的副本分片丢失该值就会增加。

number_of_pending_tasks：是指主节点创建索引并分配 shards 等任务，如果该指标数值一直未减小代表集群存在不稳定因素

active_shards_percent_as_number：集群分片健康度，活跃分片数占总分片数比例。

number_of_pending_tasks：pending task 只能由主节点来进行处理，这些任务包括创建索引并将 shards 分配给节点。

4.1.2 集群节点健康查看

```
curl http://192.168.1.51:9200/_cat/nodes?v
```

4.1.3 列出集群索引

```
curl http://192.168.1.51:9200/_cat/indices?v
```

4.1.4 在任意主机上就可以查集群状态

命令行如下：

```
curl -XGET 'http://localhost:9200/_cluster/state?pretty'
curl -XGET 'http://192.168.1.51:9200/_cluster/state?pretty'
```

4.1.5 查看模板设置信息

```
curl http://127.0.0.1:9200/_cat/templates
curl -XGET 'http://192.168.1.51:9200/_template/logstash?pretty'
```

4.1.6 关闭 elasticsearch 服务

```
curl -XPOST ‘http://localhost:9200/_shutdown’
```

4.2 kibana 操作

4.2.1 开发工具命令台常用命令

```
#查看集群健康
 GET _cluster/state?pretty
 GET _cat/health?v

# 查看节点信息
 GET _cat/nodes?v

#查看所有索引
  GET _cat/indices?v

#查看单个索引
 GET yblogs-2020.09.09/_search?Pretty

#删除索引 加*模糊匹配
 DELETE yblogs*

#查看所有模板
GET _cat/templates

#查看某个模板信息
GET _template/logstash

#设置模板信息
PUT _template/yblogs
```

4.2.2 开启堆栈监控

打开文件 config/elasticsearch.yaml 增加如下配置：

```
xpack.monitoring.collection.enabled: true

xpack.monitoring.exporters.my_local:
  type: local
  use_ingest: false
```

4.2.3 启用 x-pack 密码验证

切换到非 root 用户下，使用下面命令生成证书

./elasticsearch-certutil cert -out config/elastic-certificates.p12 -pass ""

打开文件 config/elasticsearch.yaml

xpack.security.enabled: true

xpack.security.transport.ssl.enabled: true

xpack.security.transport.ssl.verification_mode: certificate

xpack.security.transport.ssl.keystore.path: elastic-certificates.p12

xpack.security.transport.ssl.truststore.path: elastic-certificates.p12

启动 elasticsearch

./elasticsearch -d

自动生成默认用户和密码

bin/elasticsearch-setup-passwords auto

打开 config/kibana.yml 文件

elasticsearch.username: "user"

elasticsearch.password: "pass"

4.2.4 压测环境 es 自动生成的密码

开启主节点

cd /usr/share/elasticsearch/bin

//生成随机密码

elasticsearch-setup-passwords auto

or

//设置默认账号密码

elasticsearch-setup-passwords interactive

压测环境的密码

Changed password for user apm_system

PASSWORD apm_system = LVBjfLXPnUCNXb8q71sD

Changed password for user kibana_system

PASSWORD kibana_system = WgL5qLoVkTHMEe24fx5g

Changed password for user kibana

PASSWORD kibana = WgL5qLoVkTHMEe24fx5g

Changed password for user logstash_system

PASSWORD logstash_system = upYSBrNUqo4jqfY2NYyr

Changed password for user beats_system

PASSWORD beats_system = t5F5mEiNDKwj649ROGwd

Changed password for user remote_monitoring_user

PASSWORD remote_monitoring_user = sqig1w2g6UmtINaRWaN8

Changed password for user elastic

PASSWORD elastic = vcds8zeiKPoqr0eFzUqr

五 . ELK 的调优

5.1 logstash 的调优

5.1.1 logstash 消费低

logstash 从 kafka 取出数据放到 ES 中的速度一直上不去,ES 集群的负载也挺低，通过增大 logstash 的启动参数-b -w 的值，同时增大配置文件 flush_size 和 idle_flush_time 的值，速度有了较大的提升。

./bin/logstash -b 3000 -w 48 -f ./config/logstash.conf

5.1.2 logstash 吞吐量

logstash 的拉去速率太慢，吞吐量存在瓶颈，影响 es 的写入速率修改配置文件 logstash.yml。

#每次发送的事件数

pipeline.batch.size: 2048

#发送延时：

pipeline.batch.delay: 50

#监视配置文件的改变，并且当配置文件被修改以后自动重新加载配置文件

config.reload.automatic: false

5.2 为 elasticsearch 配置模板

在使用 logstash 收集日志的时候,我们一般会使用 logstash 自带的动态索引模板，虽然无须我们做任何定制操作，就能把我们的日志数据推送到

elasticsearch 索引集群中

但是在我们查询的时候，就会发现，默认的索引模板常常把我们不需要分词的字段，给分词了，这样以来，我们的比较重要的聚合统计就不准确了：所以这时候，就需要我们自定义一些索引模板了

5.2.1 模板方式

在 logstash 与 elasticsearch 集成的时候，总共有如下几种使用模板的方式：

1）使用默认自带的索引模板 ，大部分的字段都会分词，适合开发和时候快速验证使用；

2）在 logstash 收集端自定义配置模板，因为分散在收集机器上，维护比较麻烦；

3) 在 elasticsearc 服务端自定义配置模板，由 elasticsearch 负责加载模板，可动态更改，全局生效，维护比较容易。

使用 3)时，logstash 的 conf 文件需要

在 elasticsearc 服务端自定义配置模板

manage_template => false//关闭 logstash 自动管理模板功能

template_name => "xxx"//映射模板的名字

第三种需要在 elasticsearch 的集群中的 config/templates 路径下配置模板 json

4）在 kibana 配置动态模板 可以自动加载（7.x 以上）

5.2.2 模板的参数说明

观察模板的主要组成，主要有两个部分 settings 和 mappings，mappings 是用来描述数据结构的文件，而 settings 主要和索引的设置有关，这篇文章主要就是为了记录 settings 的配置项。观察别的通过文件设置的 settings 的配置，外层必包裹一个 index 目前整理的（收集到的）配置项

动态索引设置编辑

以下是与任何特定索引模块都不相关的所有动态索引设置的列表：

number_of_shards

主分片数，一般设置为数据节点的倍数

index.number_of_replicas

每个主分片具有的副本数。默认为 1。

分片在被视为搜索空闲之前不能接收搜索或获取请求的时间。（默认为 30s）

index.refresh_interval

多久执行一次刷新操作，使搜索到的索引最近更改可见。默认为 1s。可以设置 -1 为禁用刷新;设置大一点可以提高写入的性能。

best_compression

这个索引设置了 "codec":"best_compression" ，也就是官网说的压缩比更高的 DEFLATE 算法。

floor_segment

归并线程配置，segment 归并的过程，需要先读取 segment，归并计算，再写一遍 segment，最后还要保证刷到磁盘。可以说，这是一个非常消耗磁盘 IO 和 CPU 的任务。所以，ES 提供了对归并线程的限速机制，确保这个任务不会过分影响到其他任务。

5.2.3 模板的例子

```
PUT _template/logstash
{
  "order": 0,
  "index_patterns": [
    "logstash-*"
  ],
  "settings": {
    "index": {
```

```
    "number_of_shards": 12,
    "number_of_replicas" : 0,
    "codec": "best_compression",
    "refresh_interval": "180s"
  },
  "index.merge.policy.floor_segment": "100mb"
},
"mappings": {
  "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",
```

```
            "ignore_above": 256
          }
        }
      }
    }
  ],
"properties": {
    "@timestamp": {
      "type": "date"
    },
    "@version": {
      "type": "keyword"
    },
    "geoip": {
      "dynamic": true,
      "properties": {
        "ip": {
          "type": "ip"
        },
        "location": {
          "type": "geo_point"
        },
        "latitude": {
          "type": "half_float"
        },
        "longitude": {
          "type": "half_float"
        }
```

```
          }
        }
      }
  },
  "aliases": {}

}
```


六． 其他

6.1 其他的框架参考

场景：

1） datasource->logstash->elasticsearch->kibana

2） datasource->filebeat->logstash-> elasticsearch->kibana

3 ） datasource->filebeat->logstash->redis/kafka->logstash-> elasticsearch->kibana

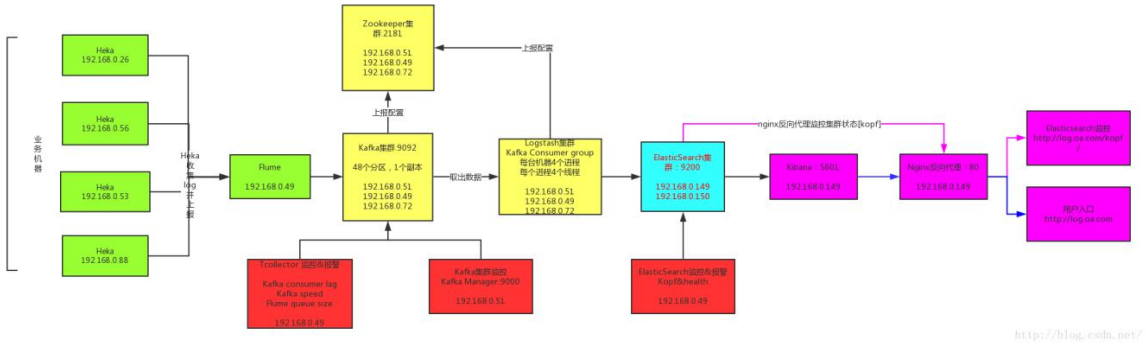
4） kafka->logstash-> elasticsearch->kibana

5） datasource->filebeat->kafka->logstash->elasticsearch->kibana(最常用)

6）datasource->logstash->redis/kafka->logstash->elasticsearch->kibana

7）mysql->logstash->elasticsearch->kibana

8）datasource->flume->kafka->logstash->elasticsearch->kibana



6.2 ELK 其他工具

6.2.1 客户端工具 cerebro 部署安装

将 cerebro 对应的安装包拷贝到服务器的指定目录，这里只需要选一个节点安装就可以了

下载地址：https://github.com/lmenezes/cerebro/releases

解压：tar -zxvf cerebro-0.8.1.tgz

在解压后的当前目录，进去配置文件 vim conf/application.conf，只需要配置如下信息：

```
hosts = [
  #{
  #  host = "http://localhost:9200"
  #  name = "Some Cluster"
  #},
  # Example of host with authentication
  {
    host = "http://master:9200"
```

```
  #  name = "Secured Cluster"
    auth = {
      username = "admin"
      password = "admin"
    }
  }
]
```

其他的可以忽略，配置好后启动,在 cerebro 当前目录执行

```
sh bin/cerebro -Dhttp.port=9001
```

注意：

启动之前 es 集群最好的启动成功状态

最好指定端口，默认端口是 9000，如果 es 集群和 hadoop 共用，则会出现端口冲突导致启动不成功

页面访问验证，地址 http://master:9001/，如果结果入下面，就是正确的

6.2.2 metricbeat 部署安装

1、下载

https://artifacts.elastic.co/downloads/beats/metricbeat/metricbeat-6.5.1-x86_64.rpm

2、安装

使用 rz 命令把安装包导入，然后执行下面的命令。

```
yum install metricbeat-6.5.1-x86_64.rpm
```

3、配置 metricbeat:

```
vim /etc/metricbeat/metricbeat.yml
```

找到下面几个配置节并修改

复制代码

```
# 直接发送 elasticsearch
output.elasticsearch:
   hosts: ["localhost:9200"]
# 要加载仪表板，可以在 metricbeat 设置中启用仪表板加载。当仪表板加载被启用时，Metricbeat 使用 Kibana API 来加载样本仪表板。只有当 Metricbeat 启动时，才会尝试仪表板加载。
# 设置 kibana 服务地址
setup.kibana:
    host: "localhost:5601"
# 加载默认的仪表盘样式
setup.dashboards.enabled: true
```

七．文献

文献

1. https://blog.csdn.net/zhousenshan/article/details/81023857?utm_medium=distribute.pc_relevant.none-task-blog-title-5&spm=1001.2101.3001.4242
2. https://bbs.huaweicloud.com/forum/thread-20661-1-1.html
3. https://www.elastic.co/cn/downloads/
4. https://www.tizi365.com/archives/793.html
5. https://blog.51cto.com/yht1990/2294103
6. https://www.jianshu.com/p/fc2aee61887f
7. https://www.elastic.co/guide/en/elasticsearch/reference/current/index-templates.html
8. https://www.cnblogs.com/zz0412/p/10573345.html
9. https://www.jianshu.com/p/532b540d4c46
10. https://www.elastic.co/cn/blog/getting-started-with-elasticsearch-security