ElasticSearch安装（集群）

1.集群规划

ip:port	节点名称	节点角色
192.168.242.139:9200，9300	es-node-1	master,data
192.168.242.140:9200，9300	es-node-2	master,data
192.168.242.141:9200，9300	es-node-3	master,data

2.环境准备

3台服务器都需要操作

把用户加入docker用户组

```
sudo gpasswd -a $USER docker
```

拉取ES-8.4.3版本镜像

```
docker pull docker.elastic.co/elasticsearch/elasticsearch:8.4.3
```

修改服务器配置文件

```
sudo vi /etc/sysctl.conf
# 文件内容为：
vm.max_map_count=262144
# 让修改生效
sudo sysctl -p
```

创建ES数据目录

```
# 创建ES文件夹
sudo mkdir -p /data/es/
# 修改es文件夹权限
sudo chmod -R 777 /data/es

mkdir -p /data/es/config
mkdir -p /data/es/data
mkdir -p /data/es/logs
mkdir -p /data/es/plugins/ik
```

先启动一次ES

在其中一台服务器执行

需要先启动一次ES，ES会自动生成SSL证书，再把证书文件拷贝出来。把生成的证书拷贝到其他ES节点。

```
# 启动一次es，生成证书及基础配置文件
docker run -d --net=host --name es-node-1
docker.elastic.co/elasticsearch/elasticsearch:8.4.3
# 等待ES启动，30秒左右。把config目录拷贝到/data/es
docker cp es-node-1:/usr/share/elasticsearch/config /data/es
# 删除本次启动的容器
docker stop es-node-1
docker rm es-node-1
# 复制配置文件到其他节点,注意修改用户名和机器ip
scp -r /data/es/config/ fan@192.168.242.140:/data/es
scp -r /data/es/config/ fan@192.168.242.141:/data/es
```

修改ES配置文件

```
cd /data/es/config
rm -rf elasticsearch.yml
vi elasticsearch.yml
```

elasticsearch.yml文件内容如下：不同机器需注意修改 node.name 和 network.host

```
cluster.name: es-cluster
node.name: es-node-1
network.host: 192.168.242.139
http.port: 9200
transport.port: 9300
# Enable security features
xpack.security.enabled: true
xpack.security.enrollment.enabled: true
# Enable encryption for HTTP API client connections, such as Kibana, Logstash,and
Agents
xpack.security.http.ssl:
  enabled: true
  keystore.path: certs/http.p12
# Enable encryption and mutual authentication between cluster nodes
xpack.security.transport.ssl:
  enabled: true
  verification_mode: certificate
  keystore.path: certs/transport.p12
  truststore.path: certs/transport.p12
# Create a new cluster with the current node only
# Additional nodes can still join the cluster later
discovery.seed_hosts: ["192.168.242.139:9300",
"192.168.242.140:9300","192.168.242.141:9300"]
cluster.initial_master_nodes: ["es-node-1", "es-node-2", "es-node-3"]
```

3.安装IK中文分词器

三台机器均需执行。

下载ik分词器8.4.3版本

```
cd /data/es/plugins/ik
wget https://github.com/medcl/elasticsearch-analysis-
ik/releases/download/v8.4.3/elasticsearch-analysis-ik-8.4.3.zip
# 安装unzip命令，已安装则跳过
sudo apt-get install unzip
unzip elasticsearch-analysis-ik-8.4.3.zip
rm -rf elasticsearch-analysis-ik-8.4.3.zip
```

4.启动ES

```
# 注意修改ES_JAVA_OPTS，推荐配置内存为服务器内存的一半
# 注意不同机器使用不同的 --name
docker run -d --restart always \
--name es-node-1 \
--net=host \
-v /data/es/data:/usr/share/elasticsearch/data \
-v /data/es/config:/usr/share/elasticsearch/config \
-v /data/es/plugins:/usr/share/elasticsearch/plugins \
-v /data/es/logs:/usr/share/elasticsearch/logs \
-e ES_JAVA_OPTS="-Xms512m -Xmx512m" \
docker.elastic.co/elasticsearch/elasticsearch:8.4.3

#如果启动失败，查看日志
docker logs es-node-1
#如果报权限错误请在执行一遍
sudo chmod -R 777 /data/es
```

下次启动使用命令：

```
docker restart es-node-1
```

安装Kibana(单机)

1.环境准备

拉取Kibana-8.4.3版本镜像

```
docker pull docker.elastic.co/kibana/kibana:8.4.3
```

创建Kibana数据目录

```
sudo mkdir -p /data/kibana/data
# 修改kibana文件夹权限
sudo chmod -R 777 /data/kibana

mkdir -p /data/kibana/data
mkdir -p /data/kibana/config
mkdir -p /data/kibana/logs
```

创建Kibana配置文件

```
cd /data/kibana/config
vi kibana.yml
# 文件内容：
server.port: 5601
server.host: "0.0.0.0"
i18n.locale: "zh-CN"
```

2.启动Kibana

首次启动：

```
docker run -it -d --restart always -p 5601:5601 \
--name kibana \
-v /data/kibana/data:/usr/share/kibana/data \
-v /data/kibana/config:/usr/share/kibana/config \
-v /data/kibana/logs:/usr/share/kibana/logs \
docker.elastic.co/kibana/kibana:8.4.3
```

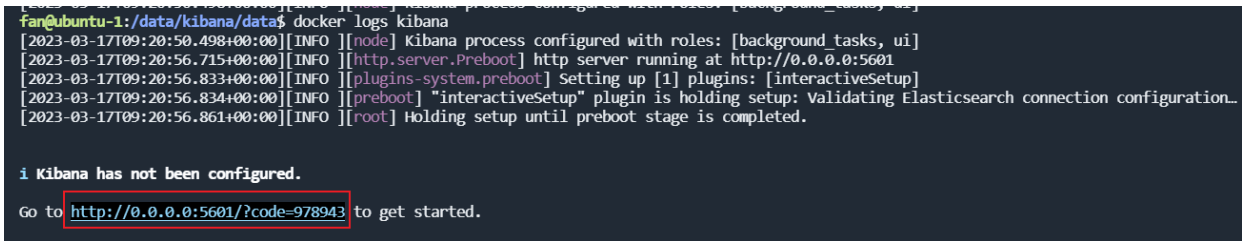
下次启动使用命令：

```
docker restart kibana
```

3.Kibana连接ES

启动kibana后，执行以下命令可以看到kibana访问code

```
docker logs kibana
```



在浏览器访问上面url进入kibana首页，上面的0.0.0.0替换成kibana主机服务器ip

进入es-node-1，获取Kibana令牌，修改ES密码

```
# 修改ES管理员账户elastic 密码为 odcsz@1236547890
docker exec -it es-node-1 /usr/share/elasticsearch/bin/elasticsearch-reset-password
-u elastic -i
# 生成kibana Token
docker exec -it es-node-1 /usr/share/elasticsearch/bin/elasticsearch-create-
enrollment-token -s kibana
```

得到token，并将token粘贴进第2步的kibana页面中，kibana将自动完成配置

eyJ2ZXIiOiI4LjQuMyIsImFkciI6WyIxNzIuMTcuMC4yOjkyMDAiXSwiZmdyIjoiMGNiYzQzYWIyZDI4NGZmZmQzYzBjNDVkMzUyOGY3YmViOWQ1Y2IwZGMyMWY2ZDUzM2JjZTMwNDUwOWUyZDhhOSIsImtleSI6ImF1cm03b1lCem1uMXlVSEVlSktlOnFZYXRKVnI3UktTd2VSSmpyV3NiRkEifQ==

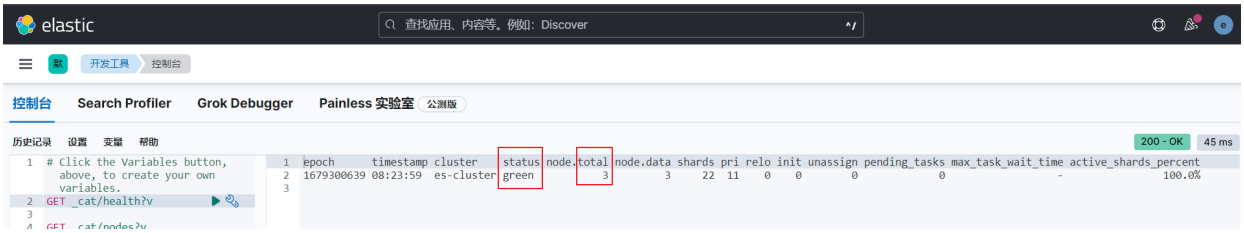
如果这里需要输入授权码，使用docker logs kibana 查看即可。

如果报权限错误，无法写入kibana.yml，再执行一遍 sudo chmod -R 777 /data/kibana

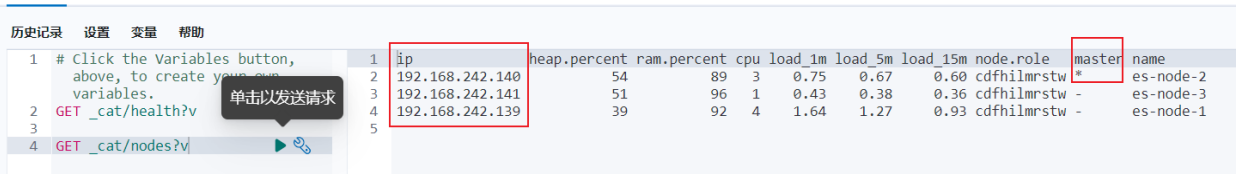
使用 elastic/odcsz@1236547890 登录即可进入Kibana首页。

验证配置成功

在Kibana开发工具执行：GET _cat/health?v 发现集群状态为 green 集群节点为3



在Kibana控制台执行：GET _cat/nodes?v 可以查看各节点ip，以及主节点所在机器



验证配置成功

安装Filebeat

filebeat支持采集本地日志文件内容传送到es集群。每台需要采集日志的服务器均需要启动一个filebeat实例。

1.环境准备

拉取Filebeat-8.4.3版本镜像

```
docker pull docker.elastic.co/beats/filebeat:8.4.3
```

创建Filebeat目录

```
sudo mkdir -p /data/filebeat
sudo chmod -R 777 /data/filebeat
mkdir -p /data/filebeat/data
```

创建配置文件

```
cd /data/filebeat
vi filebeat.yml
```

```
# 日志路径配置、ES集群ip配置以实际生产日志为准
# 文件内容需要删除所有行尾注释
filebeat.inputs:
- type: filestream                          #读取类型为文件的日志
  id: otc-option-admin                      #文件流id必须唯一
  enabled: true
  encoding: utf-8                           #文件编码
  paths:                                    #文件路径,当此文件被重命名时,filebeat会继续监控重
命名后的文件5分钟。
    - /data/logs/otc-option-admin.log       #filebeat每10秒钟会扫描一次新生成的文件。

  fields:
    tag: 'otc-option-admin'                 #将此文件添加一个标签字段，值为'otc-option-
admin'
  parsers:
    - multiline:                            #处理多行日志，把不以20开头的行合并到上一行末尾
        type: pattern
        pattern: '^20'
```

```
        negate: true
        match: after
- type: filestream
  id: otc-option-product
  enabled: true
  encoding: utf-8
  paths:
    - /data/logs/otc-option-product.log
  fields:
    tag: 'otc-option-product'
  parsers:
    - multiline:
        type: pattern
        pattern: '^20'
        negate: true
        match: after
filebeat.config.modules:
  path: ${path.config}/modules.d/*.yml
  reload.enabled: false
setup.template.settings:
  index.number_of_shards: 3                    #索引分片数
name: 192.168.242.139                          #本filebeat实例名,这里设置为ip
output.elasticsearch:
  hosts: ["https://192.168.242.140:9200"]      #es ip:port 用户名密码
  username: "elastic"
  password: "odcsz@1236547890"
  ssl.certificate_authorities: /usr/share/filebaat/http_ca.crt  #用于连接es的证书
  pipeline: "log_pipeline"                     #用于处理日志内容的es log_pipeline
  indices:
    - index: "dev-otc-option-admin-%{+yyyy-MM-dd}"
      when.contains:
        fields:                                #把包含"otc-option-admin"这个tag的日志放
到
          tag: "otc-option-admin"              #"dev-otc-option-admin-%{+yyyy-MM-dd}"
索引里。一天生成一个索引
    - index: "dev-otc-option-product-%{+yyyy-MM-dd}"
      when.contains:
        fields:
          tag: "otc-option-product"
processors:
  - add_host_metadata:
      when.not.contains.tags: forwarded
  - add_cloud_metadata: ~
  - add_docker_metadata: ~
  - add_kubernetes_metadata: ~
```

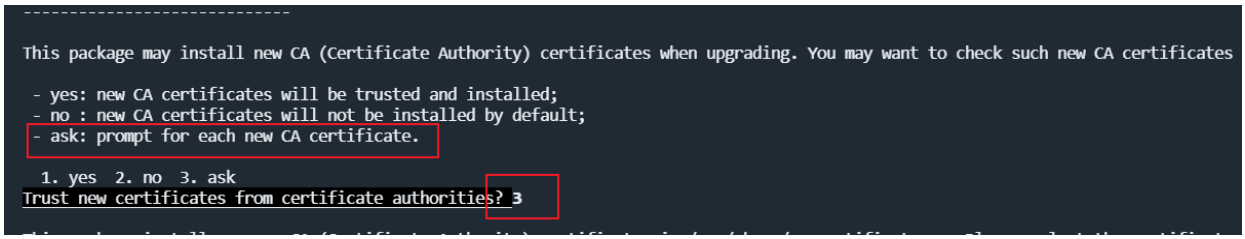
```
# 修改配置文件权限、修改所属用户为root
chmod go-w filebeat.yml
sudo chown root:root filebeat.yml
```

准备es证书

因为ES采用https访问，我们需要在filebeta所在服务器安装ES的证书才能访问ES集群。

拷贝es服务器 /data/es/config/certs/http_ca.crt 到filebeat服务器

```
# 把证书拷贝到这里
sudo cp http_ca.crt /data/filebeat
```



出现 1 added 表示安装成功。

配置es pipeline

filebeat读取日志文件，默认生成的每行日志内容（源数据）包含以下多个字段：

```
message 、host 、agent 、 ecs 、log 、 input 、 fields
```

其中`message`字段为filebeat读取到的日志文件的每一行内容，是我们需要解析的字段。

其他字段包含机器信息、文件信息、附加字段（fields）等。

在Kibana Dev Tools里配置es pipeline：

```
PUT _ingest/pipeline/log_pipeline
{
  "description": "log_pipeline",
  "processors": [
    {
      "grok": {
        "field": "message",
        "patterns": [
```

```
          """%{TIMESTAMP_ISO8601:logTime}\|%{LOGLEVEL:logLevel}\|(?
<projectName>\S*)\|(?<method>\S*)\|%{IP:clientIp}\|(?<operator>\S*)\|\[%
{DATA:traceId}\](?m)(?<msg>.*|\s)"""
        ]
      }
    },
    {
     "date": {
      "field": "logTime",
      "timezone": "Asia/Shanghai",
      "formats": ["yyyy-MM-dd HH:mm:ss,SSS"],
      "ignore_failure": true
     }
    },
    {
      "set" : {
        "field": "serverIp",
        "value": "{{{host.name}}}"
      }
    },
    {
      "remove" : {
        "field" : "agent"
      }
    },
        {
      "remove" : {
        "field" : "host"
      }
    },
            {
      "remove" : {
        "field" : "log"
      }
    },
    {
      "remove" : {
        "field" : "ecs"
      }
    },
    {
      "remove" : {
        "field" : "input"
      }
    },
        {
      "remove" : {
        "field" : "message"
      }
    },
    {
      "remove" : {
```

```
        "field" : "fields"
      }
    },
        {
      "remove" : {
        "field" : "logTime"
      }
    }
  ]
}
```

配置说明：

`grok`：使用grok语法解析`源数据`的`message`字段。根据日志格式，使用grok表达式：

```
%{TIMESTAMP_ISO8601:logTime}\|%{LOGLEVEL:logLevel}\|(?<projectName>\S*)\|(?
<method>\S*)\|%{IP:clientIp}\|\[%{DATA:traceId}\](?m)(?<msg>.*|\s)
```

解析可以生成以下几个目标字段：

logTime 、 logLevel 、 projectName 、method、clientIp、traceId、msg

`date`: es默认生成的`@timestamp`字段是insert data的时间，并不是日志里的时间。

配置`date`解析`logTime`字段替换`@timestamp`字段。方便按日志时间检索、做数据看板。

`set`:添加一个目标字段

`remove`:移除一些没用的`源数据`里的字段。

2.启动Filebeat

创建容器并启动

```
docker run -d --restart always \
--name filebeat \
--user root \
-v /data/filebeat/filebeat.yml:/usr/share/filebeat/filebeat.yml \
-v /data/filebeat/data:/usr/share/filebeat/data \
-v /data/filebeat/http_ca.crt:/usr/share/filebeat/http_ca.crt \
-v /data/logs:/data/logs \  # 注意修改这里的日志文件路径
docker.elastic.co/beats/filebeat:8.4.3
```

下次启动

```
docker restart filebeat
```