前言

ELK是Elasticsearch、Logstash、Kibana的简称，这三者是核心套件，但并非全部。

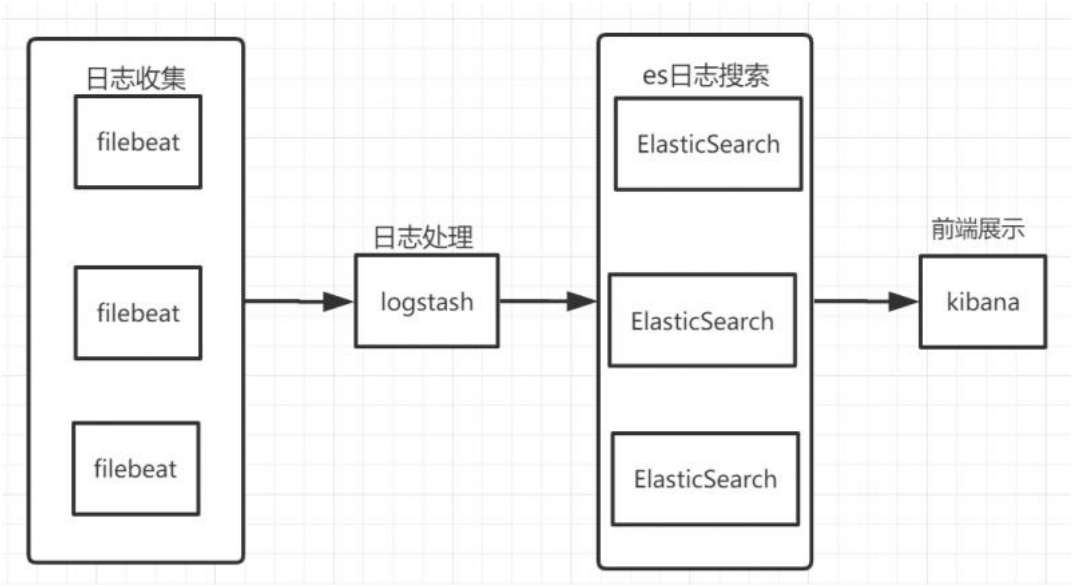
Elasticsearch是实时全文搜索和分析引擎，提供搜集、分析、存储数据三大功能；是一套开放REST和JAVA API等结构提供高效搜索功能，可扩展的分布式系统。它构建于Apache Lucene搜索引擎库之上。

Logstash是一个用来搜集、分析、过滤日志的工具。它支持几乎任何类型的日志，包括系统日志、错误日志和自定义应用程序日志。它可以从许多来源接收日志，这些来源包括 syslog、消息传递（例如 RabbitMQ）和JMX，它能够以多种方式输出数据，包括电子邮件、websockets和Elasticsearch。

Kibana是一个基于Web的图形界面，用于搜索、分析和可视化存储在 Elasticsearch指标中的日志数据。它利用Elasticsearch的REST接口来检索数据，不仅允许用户创建他们自己的数据的定制仪表板视图，还允许他们以特殊的方式查询和过滤数据。

Filebeat是本地文件的日志数据采集器。 作为服务器上的代理安装，Filebeat监视日志目录或特定日志文件，tail file，并将它们转发给Elasticsearch或Logstash进行索引、kafka 等。

linux本地docker.log--》filebeat（收集日志）--》logstash（过滤）--》elasticsearch（添加索引）--》kibana检索显示



docker-elk 服务端 ：git@github.com:jiangxd0716/ELK-filebeat.git　　elasticsearch logstash kibana

elk-filebeat 客户端：git@github.com:jiangxd0716/ELK-filebeat.git　　filebeat用于收集日志，传给elk

环境

```
Centos7
192.168.8.20  elasticsearch logstash kibana filebeat
192.168.8.10  filebeat
192.168.8.30  filebeat
```

服务：

- elasticsearch：9200
- logstash：5000
- kibana：5601

安装

一、安装docker-elk服务端

1.拉取代码

- git clone git@github.com:jiangxd0716/ELK-filebeat.git

2.安装docker及docker-compose

参考：https://www.cnblogs.com/jxd283465/p/11542127.html

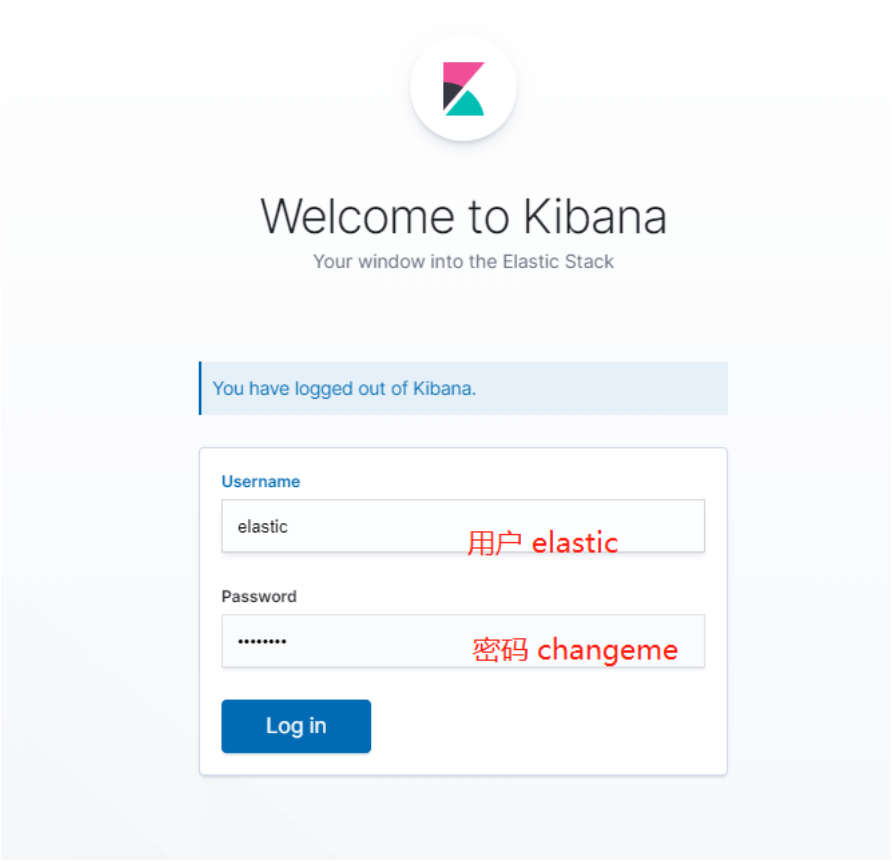
3.启动服务端

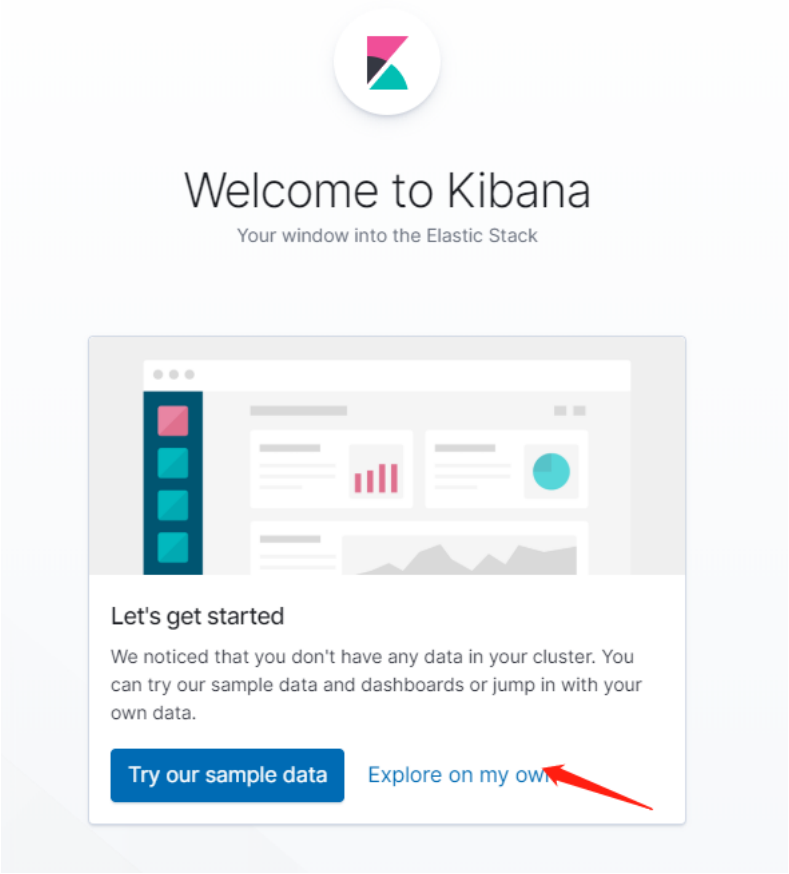
```
[root@localhost docker-elk]# pwd
/home/ELK-filebeat/docker-elk
[root@localhost docker-elk]# docker-compose up -d
Starting dockerelk_elasticsearch_1 ...
```

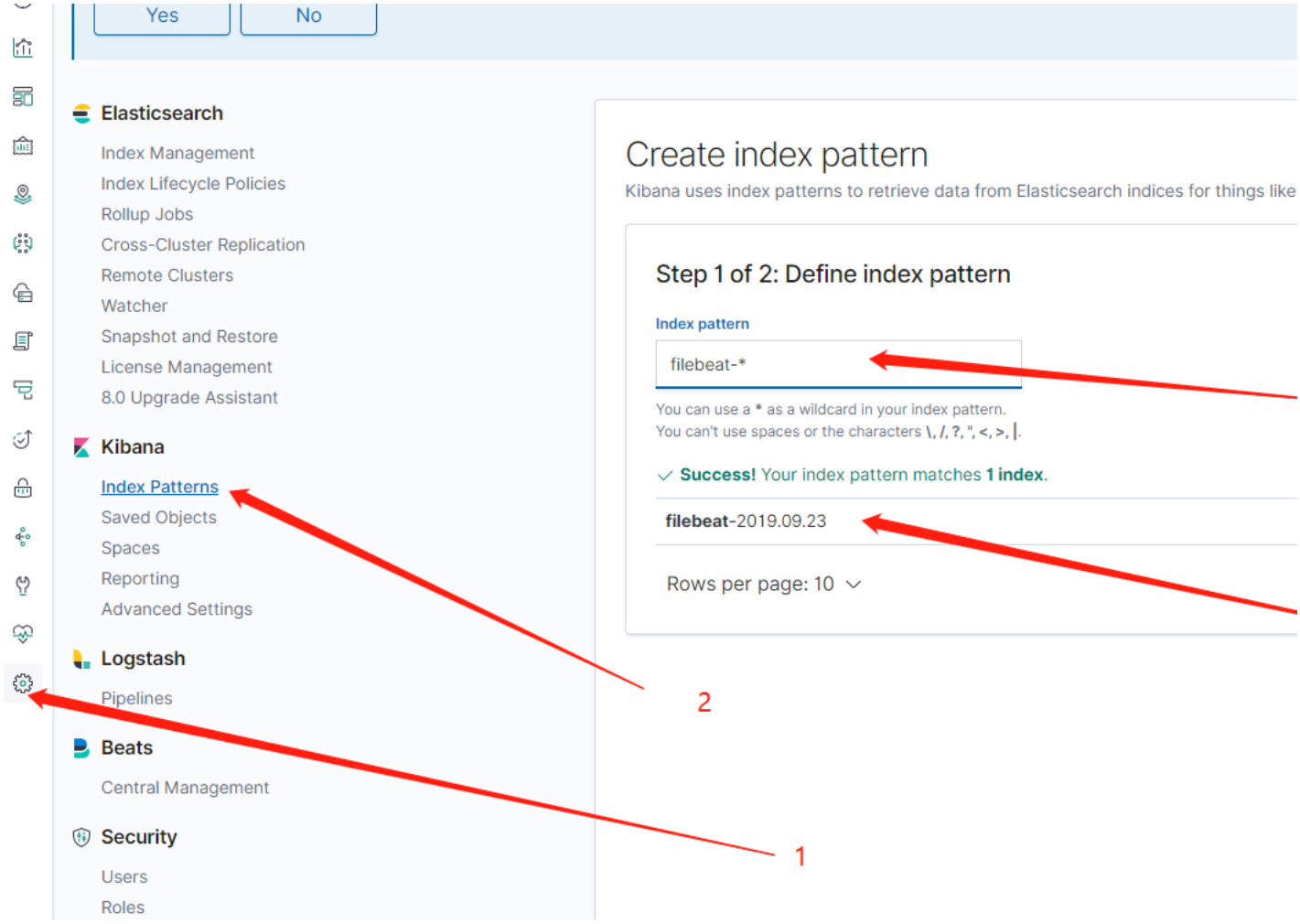
```
Starting dockerelk_elasticsearch_1 ... done
Starting dockerelk_kibana_1 ...
Starting dockerelk_logstash_1 ...
Starting dockerelk_kibana_1
Starting dockerelk_kibana_1 ... done
[root@localhost docker-elk]# docker ps
CONTAINER ID        IMAGE                  COMMAND                  CREATED             STATUS              PORTS
7152edaa71dc        dockerelk_kibana       "/usr/local/bin/dumb…"   3 minutes ago       Up 4 seconds        0.0.0.0:5601->5601/tcp
63908878fb00        dockerelk_logstash     "/usr/local/bin/dock…"   3 minutes ago       Up 4 seconds        0.0.0.0:5000->5000/tcp, 0.0.0.0:9600->9600/tcp, 5044
d3c44ac2264c        dockerelk_elasticsearch "/usr/local/bin/dock…"   3 minutes ago       Up 4 seconds        0.0.0.0:9200->9200/tcp, 0.0.0.0:9300->9300/tcp
```

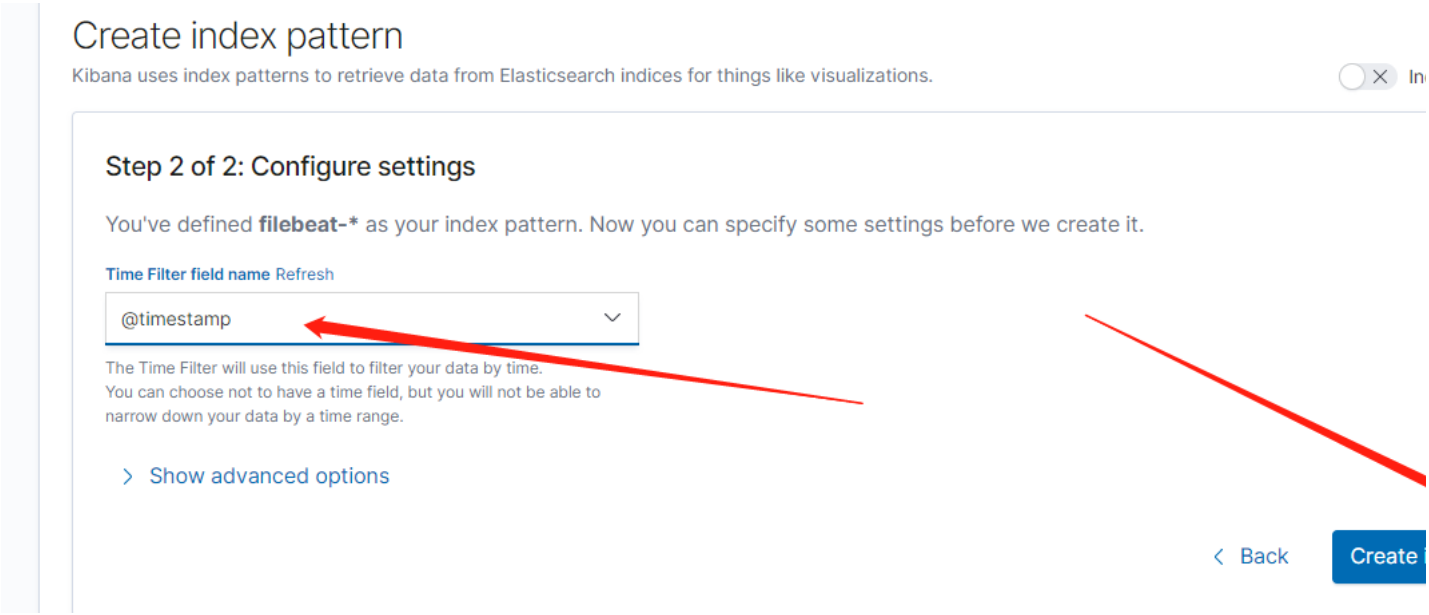
4.访问kibana

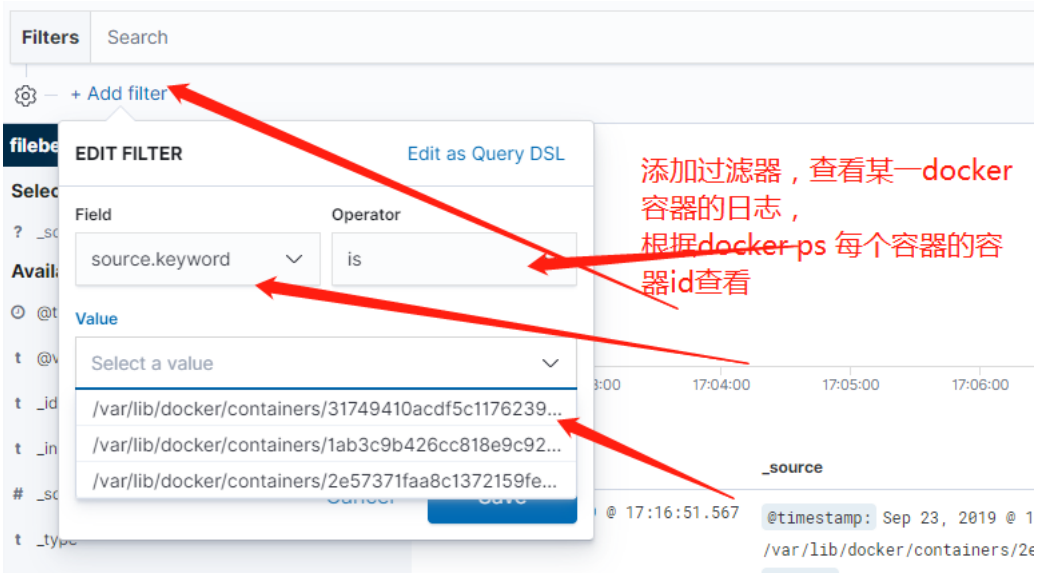
浏览器访问 http://192.168.8.20:5601






Yes
No
Elasticsearch
Index Management
Index Lifecycle Policies
Rollup Jobs
Cross-Cluster Replication
Remote Clusters
Watcher
Snapshot and Restore
License Management
8.0 Upgrade Assistant
Kibana
Index Patterns
Saved Objects
Spaces
Reporting
Advanced Settings
Logstash
Pipelines
Beats
Central Management
Security
Users
Roles
Create index pattern
Kibana uses index patterns to retrieve data from Elasticsearch indices for things like
Step 1 of 2: Define index pattern
Index pattern
filebeat-*
You can use a * as a wildcard in your index pattern.
You can't use spaces or the characters \, /, ?, ", <, >, |.
Success! Your index pattern matches 1 index.
filebeat-2019.09.23
Rows per page: 10
2
1


Create index pattern
Kibana uses index patterns to retrieve data from Elasticsearch indices for things like visualizations.
Step 2 of 2: Configure settings
You've defined filebeat-* as your index pattern. Now you can specify some settings before we create it.
Time Filter field name Refresh
@timestamp
The Time Filter will use this field to filter your data by time.
You can choose not to have a time field, but you will not be able to
narrow down your data by a time range.
Show advanced options
Back
Create



二、安装客户端

1.修改filebeat的配置文件filebeat.yml-------output到elasticsearch或者logstash

```
[root@localhost config]# pwd
/home/ELK/ELK/docker-elk/filebeat/config
[root@localhost config]# vi filebeat.yml
[root@localhost config]# cat filebeat.yml
filebeat.prospectors:
- type: log
  enabled: true
  paths:
    - /var/lib/docker/containers/*/*.log #需要读取日志的目录#这里读取的是docker的日志文件，也可以加入tomcat、nginx等配置文件
  document_type: syslog
  json.keys_under_root: true # 因为docker使用的log driver是json-file，因此采集到的日志格式是json格式，设置为true之后，filebeat会将日志进行json_decode处理
  json.add_error_key: true #如果启用此设置，则在出现JSON解组错误或配置中定义了message_key但无法使用的情况下，Filebeat将添加“error.message”和“error.type：json”键
  json.message_key: log #一个可选的配置设置，用于指定应用行筛选和多行设置的JSON密钥。 如果指定，键必须位于JSON对象的顶层，且与键关联的值必须是字符串，否则不会发生过
  tail_files: true
  # 将error日志合并到一行
  multiline.pattern: '^([0-9]{4}|[0-9]{2})-[0-9]{2}'
  multiline.negate: true
  multiline.match: after
  multiline.timeout: 10s
#  registry_file: /opt/filebeat/registry
#------------------------- Elasticsearch output ------------------------------
# 直接输出到elasticsearch,这里的hosts是elk地址，端口号是elasticsearch端口#
#output.elasticsearch:
#  hosts: ["192.168.8.100:9200"]
#  username: "elastic"
#  password: "changeme"
output:
 logstash:
  enabled: true
  hosts:
     - 192.168.8.20:5000  #这里将filebeat收集的日志output到logstash，此处为logstash的ip和端口，也可以直接输送到elasticsearch
#================== Elasticsearch template setting ==========================
setup.template.name: "filebeat.template.json"
setup.template.fields: "filebeat.template.json"
setup.template.overwrite: true
setup.template.enabled: false
# 过滤掉一些不必要字段#
processors:
- drop_fields:
    fields: ["input_type", "offset", "stream", "beat"]
```

2.启动客户端

```
[root@localhost elk-filebeat]# pwd
/home/ELK-filebeat/elk-filebeat
[root@localhost elk-filebeat]# docker-compose up -d
Creating network "elkfilebeat_default" with the default driver
Creating elkfilebeat_filebeat_1 ...
Creating elkfilebeat_filebeat_1 ... done


[root@localhost elk-filebeat]# docker ps
CONTAINER ID        IMAGE                    COMMAND                  CREATED             STATUS              PORTS
31749410acdf        prima/filebeat           "/docker-entrypoint.…"   5 minutes ago       Up 5 minutes
2e57371faa8c        dockerelk_kibana         "/usr/local/bin/dumb…"   7 minutes ago       Up 7 minutes        0.0.0.0:5601->5601/tcp
09220beadd39        dockerelk_logstash       "/usr/local/bin/dock…"   7 minutes ago       Up 7 minutes        0.0.0.0:5000->5000/tcp, 0.0.0.0:9600->9600/tcp, 5044
1ab3c9b426cc        dockerelk_elasticsearch  "/usr/local/bin/dock…"   7 minutes ago       Up 7 minutes        0.0.0.0:9200->9200/tcp, 0.0.0.0:9300->9300/tcp
```