使用docker-compose部署 ELK全家桶

前情提要：

ELK 是一个由 Elasticsearch、Logstash 和 Kibana 组成的技术栈，用于集中式日志管理和数据分析。它的主要作用包括：

1. 日志收集和处理
Logstash：作为数据收集和处理引擎，Logstash 能够从多个来源（如日志文件、数据库、消息队列等）收集数据，进行过滤、解析、增强并将其传输到 Elasticsearch。Logstash 支持丰富的插件，可以根据需要灵活配置数据管道。
2. 数据存储和搜索
Elasticsearch：一个分布式搜索和分析引擎，擅长处理大规模的数据。Elasticsearch 使用文档存储数据，能够提供近乎实时的搜索和分析能力。它的分布式架构使其具备高可用性和水平扩展能力。
3. 数据可视化和分析
Kibana：作为一个数据可视化和分析平台，Kibana 提供了丰富的图表和仪表板，用户可以通过 Kibana 直观地展示和分析存储在 Elasticsearch 中的数据。Kibana 提供了强大的过滤、查询和仪表板共享功能，适用于多种数据分析场景。
主要用途
集中式日志管理
收集和存储来自不同应用程序、服务器和网络设备的日志数据，提供统一的日志访问和管理界面。
实时监控和报警
实时监控系统性能和应用程序行为，通过创建仪表板和设置报警规则，及时发现和响应问题。
安全信息和事件管理 (SIEM)
集中收集和分析安全日志，检测和响应安全事件，满足合规性要求。
业务分析
从业务数据中提取有价值的信息，通过数据分析和可视化，支持业务决策。

ELK文件配置准备

mark:3024:0:0:0:q75.awebp#?w=414&h=116&s=6604&e=png&b=000000
)

目录层次图

![image-20240703173655865.png](https://p9-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/865c6ded091a43d4aa0eeb00c8fa7fb9~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=714&h=486&s=30820&e=png&b=ffffff)

logstash.yml

```
config:
  reload:
    automatic: true
    interval: 3s
xpack:
  management.enabled: false
  monitoring.enabled: false
```

pipelines.yml

```
- pipeline.id: logstash_dev
  path.config: /usr/share/logstash/pipeline/logstash_dev.conf
```

logstash_dev.conf

```
input {
```

```
  beats {
    port => 9900
  }
}

filter {
  grok {
    match => { "message" => "%{COMBINEDAPACHELOG}" }
  }

  mutate {
    convert => {
      "bytes" => "integer"
    }
  }

  geoip {
    source => "clientip"
  }

  useragent {
    source => "user_agent"
    target => "useragent"
  }

  date {
    match => ["timestamp", "dd/MMM/yyyy:HH:mm:ss Z"]
  }
}

output {
  stdout { }

  elasticsearch {
    hosts => ["elasticsearch:9200"]
    index => "logstash-poc"
  }
}
```

kibana.yml

```
erver.name: kibana
```

```
server.host: "0"
elasticsearch.hosts: ["elasticsearch:9200"]
xpack.monitoring.ui.container.elasticsearch.enabled: true

```

filebeat.yml

```
filebeat.inputs:
- type: log
  enabled: true
  paths:
    - /usr/share/filebeat/logs/*

output:
  logstash:
    hosts: ["logstash:9900"]

```

docker-compose.yml配置(根据自己的路径修改 volumes 挂载的信息)

```
version: '3.7'

services:
  elasticsearch:
    image: docker.elastic.co/elasticsearch/elasticsearch:7.7.1
    container_name: elasticsearch
    environment:
      - discovery.type=single-node
    ports:
      - "9201:9200"
      - "9301:9300"
    volumes:
      - /data/docker-data/elk/es/data:/usr/share/elasticsearch/data
      - /data/docker-data/elk/es/logs:/usr/share/elasticsearch/logs
      - /data/docker-
data/elk/es/config/elasticsearch.yml:/usr/share/elasticsearch/config/el
asticsearch.yml
```

```
  logstash:
    image: docker.elastic.co/logstash/logstash:7.7.1
    container_name: logstash
    ports:
      - ”5044:5044”
      - ”9600:9600”
    volumes:
      - /data/docker-
data/elk/logstash/pipeline:/usr/share/logstash/pipeline
      - /data/docker-
data/elk/logstash/config:/usr/share/logstash/config
    depends_on:
      - elasticsearch

  kibana:
    image: docker.elastic.co/kibana/kibana:7.7.1
    container_name: kibana
    ports:
      - ”5601:5601”
    environment:
      - SERVER_NAME=kibana
      - SERVER_HOST=0.0.0.0
      - ELASTICSEARCH_HOSTS=http://elasticsearch:9200
      -
XPACK_MONITORING_UI_CONTAINER_ELASTICSEARCH_ENABLED=true
    depends_on:
      - elasticsearch

  filebeat:
    image: docker.elastic.co/beats/filebeat:7.7.1
    container_name: filebeat
    volumes:
      - /data/docker-
data/elk/filebeat/filebeat.yml:/usr/share/filebeat/filebeat.yml
      - /data/docker-data/elk/filebeat/logs:/usr/share/filebeat/logs
      - /var/lib/docker/containers:/var/lib/docker/containers
    command: [”--strict.perms=false”]
    depends_on:
      - logstash
      - elasticsearch

```

下载docker-compose

```
yum install -y python3-pip
sudo pip3 install docker-compose -i
https://mirrors.aliyun.com/pypi/simple/
```

备注：需要在docker-compose.yml文件所在目录执行

启动docker-compose

```
docker-compose up -d
```

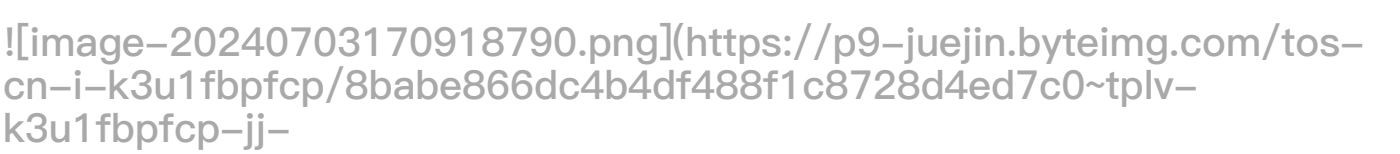
关闭docker-compose

```
docker-compose down
```

查看日志（-f 后面可以接其他container_name）

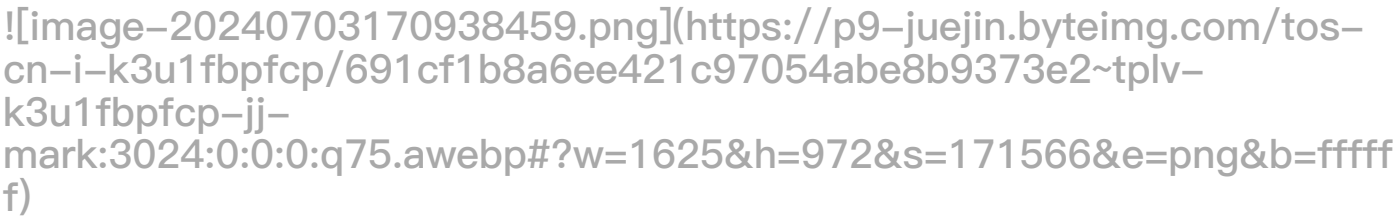
```
docker-compose logs -f elasticsearch
```

测试es启动 服务器ip:9201（端口可以自己配置）



mark:3024:0:0:0:q75.awebp#?w=514&h=432&s=17984&e=png&b=fdfdfd
)

测试kibana启动 服务器ip:5601 （容器启动后需要时间运行，稍等片刻）



测试日志解析（未接入项目，手动导入日志文件测试）

使用 Docker 进行目录创建和日志文件准备：

```
docker exec -it filebeat mkdir -p /usr/share/filebeat/logs
```

将准备好的日志文件传到filebeat容器中

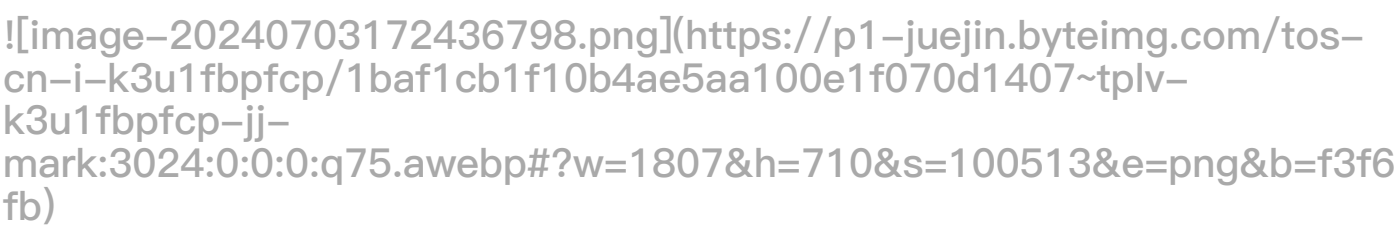
```
docker cp [your_path]/logfile.log filebeat:/usr/share/filebeat/logs/
```

重启filebeat

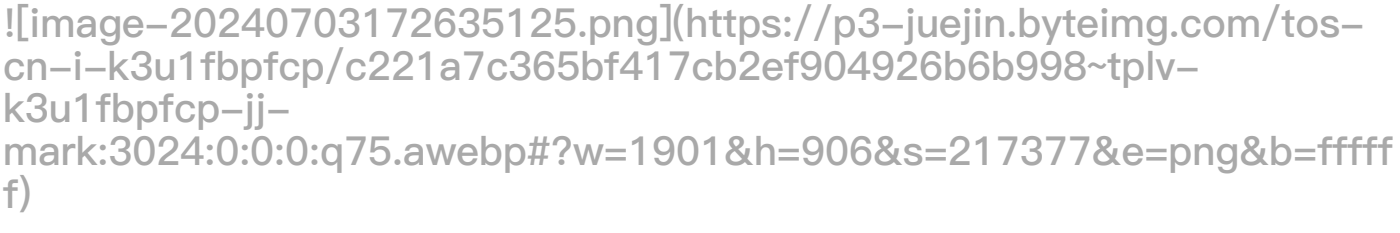
```
docker-compose restart filebeat
```

查看日志

一、创建index



二、过滤查询



结语：大家可以在此基础上根据需要配置的更灵活些。

原文链接: https://juejin.cn/post/7387250487621910537