**准备工作**

1. 开始之前，我们需要删除之前安装容器kong（不会就去翻docker教程）
2. 创建日志目录，我们创建一个目录存放日志

# mkdir -p /var/kong/logs && chmod 777 /var/kong/logs

1. 执行如下命令生成本篇教程的docker

# docker run -d --name kong \

--network kongnet \

--restart=always \

-e "KONG\_DATABASE=postgres" \

-e "KONG\_PG\_HOST=postgres" \

-e "KONG\_PG\_USER=kong" \

-e "KONG\_PG\_PORT=5432" \

-e "KONG\_PG\_PASSWORD=kong123" \

-e "KONG\_PG\_DATABASE=kong" \

-e "KONG\_PREFIX=/usr/local/kong/" \

-e "KONG\_PROXY\_ACCESS\_LOG=logs/kong\_client\_access.log" \

-e "KONG\_ADMIN\_ACCESS\_LOG=logs/kong\_admin\_access.log" \

-e "KONG\_PROXY\_ERROR\_LOG=logs/kong\_error.log" \

-e "KONG\_ADMIN\_ERROR\_LOG=logs/kong\_error.log" \

-e "KONG\_ADMIN\_LISTEN=0.0.0.0:8001" \

-e "KONG\_ADMIN\_LISTEN\_SSL=0.0.0.0:8444" \

-v /var/kong/logs:/usr/local/kong/logs \

-p 80:8000 \

-p 443:8443 \

-p 8001:8001 \

-p 8444:8444 \

kong:2.0

参数说明：上篇文章已做说明，这里不讲了

**日志相关的配置**

prefix：工作目录，包含临时文件和日志

log\_level：日志等级，取值为debug、info、notice、warn、error等

proxy\_access\_log：代理请求的日志路径，设为 off 可以关闭

proxy\_error\_log：代理错误的日志路径

admin\_access\_log：Admin Api请求访问日志路径，设为 off 可以关闭

admin\_error\_log：Admin Api请求错误日志路径

**查看日志**

Kong的日志被我们映射到了/var/kong/logs目录

# ls /var/kong/logs

**日志格式**

如下是默认的日志格式

172.18.0.1 - - [04/Jan/2022:13:26:42 +0000] "GET /netkongtest/WeatherForecast HTTP/1.1" 200 513 "-" "Mozilla/5.0 (X11; Ubuntu; Linux x86\_64; rv:95.0) Gecko/20100101 Firefox/95.0"

172.18.0.1 - - [04/Jan/2022:13:26:44 +0000] "GET /netkongtest/WeatherForecast HTTP/1.1" 200 507 "-" "Mozilla/5.0 (X11; Ubuntu; Linux x86\_64; rv:95.0) Gecko/20100101 Firefox/95.0"

172.18.0.1 - - [04/Jan/2022:13:26:45 +0000] "GET /netkongtest/WeatherForecast HTTP/1.1" 200 510 "-" "Mozilla/5.0 (X11; Ubuntu; Linux x86\_64; rv:95.0) Gecko/20100101 Firefox/95.0"

说明：

172.18.0.1：访问者ip

[04/Jan/2022:13:26:42 +0000]：访问时间

"GET /netkongtest/WeatherForecast HTTP/1.1"：访问的Api

"Mozilla/5.0 ..."：浏览器信息

**分布式日志**

分布式系统的日志主要的问题是如何将各个系统的日志收集到一个地方，好进行查看和分析，下面介绍几个服务用于收集、处理、和检索日志

**Filebeat**

Filebeat是一个日志采集器，其收集日志并将其发送到Elasticsearch或Logstash中。

如下，我们搭建Filebeat

1. 拉取镜像

# docker pull elastic/filebeat:7.5.0

1. 编写filebeat的配置文件

如下，我们在/opt/filebeat/config/目录下创建filebeat.yml文件，其内容如下：

# ------------------------------ input -------------------------------

filebeat.inputs:

#input类型，默认是log

- type: log

#是否启用该type

enabled: true

#input文件的位置，可以有多个

paths:

- /var/kong/logs/kong\_client\_access.log

- /var/kong/logs/kong\_error.log

# ------------------------------ Logstash Output -------------------------------

#output到logstash

output.logstash:

#logstash服务器地址，可以有多个，以逗号","隔开

hosts: ["logstash:5044"]

其中logstash:5044是logstash服务器的地址，稍后我们将创建该服务器（docker容器）

1. 执行如下命令生成docker容器

# docker run -d --name filebeat \

--network kongnet \

--restart=always \

-v /opt/filebeat/config/filebeat.yml:/usr/share/filebeat/filebeat.yml \

-v /var/kong/logs:/var/kong/logs \

elastic/filebeat:7.5.0

说明：

/usr/share/filebeat/filebeat.yml：filebeat默认配置文件

/var/kong/logs：将本地的/var/kong/logs映射到容器的/var/kong/logs

**Logstash**

logstash是一个日志处理器，这里我们接收来自filebeat的日志，处理后发送至Elasticsearch（实际上这里的示例我们并不做任何处理）

1. 拉起容器

#docker pull logstash:7.5.0

1. 创建/opt/logstash/logstash.conf配置文件，其内容如下

input {

# 接收来自 filebeat 的数据，暴露的端口为 5044

beats {

port => 5044

}

}

output {

# 将处理的数据发送至 elasticsearch

elasticsearch {

hosts => ["elasticsearch:9200"]

index => "filebeat-logstash-cloud-kong-access"

}

}

其中elasticsearch:9200是elasticsearch服务器的地址，稍后我们将创建该容器

1. 运行容器

# docker run -d --name logstash \

--network kongnet \

--restart=always \

-p 5044:5044 \

-v /opt/logstash/logstash.conf:/usr/share/logstash/pipeline/logstash.conf \

logstash:7.5.0

**Elasticsearch**

elasticsearch是一个文本检索引擎，我们这里用于检索我们的日志

1. 拉起容器

# docker pull elasticsearch:7.5.0

1. 创建elasticsearch数据存储目录

# mkdir -p /opt/elasticsearch/data && chmod 777 /opt/elasticsearch/data

1. 创建配置文件 /opt/elasticsearch/config/elasticsearch.yml，其内容如下：

network.bind\_host: 0.0.0.0

http.cors.enabled: true

http.cors.allow-origin: "\*"

1. 运行容器

# docker run -d --name elasticsearch \

--network kongnet \

--restart=always \

-e "discovery.type=single-node" \

-p 9200:9200 \

-p 9300:9300 \

-v /opt/elasticsearch/data:/usr/share/elasticsearch/data \

-v /opt/elasticsearch/config/elasticsearch.yml:/usr/share/elasticsearch/config/elasticsearch.yml \

elasticsearch:7.5.0

至此我们分布式日志处理完成，可以通过elasticsearch的api检索日志

**Kibana**

kibana是针对elasticsearch的可视化站点，下面我们安装kibana，使用kibana检索收集过来的日志

1. 拉取容器

# docker pull kibana:7.5.0

1. 创建配置文件/opt/kibana/config/kibana.yml，其内容如下：

server.name: kibana

server.host: "0.0.0.0"

elasticsearch.hosts: ["http://elasticsearch:9200"]

1. 运行容器

# docker run -d --name kibana \

--network kongnet \

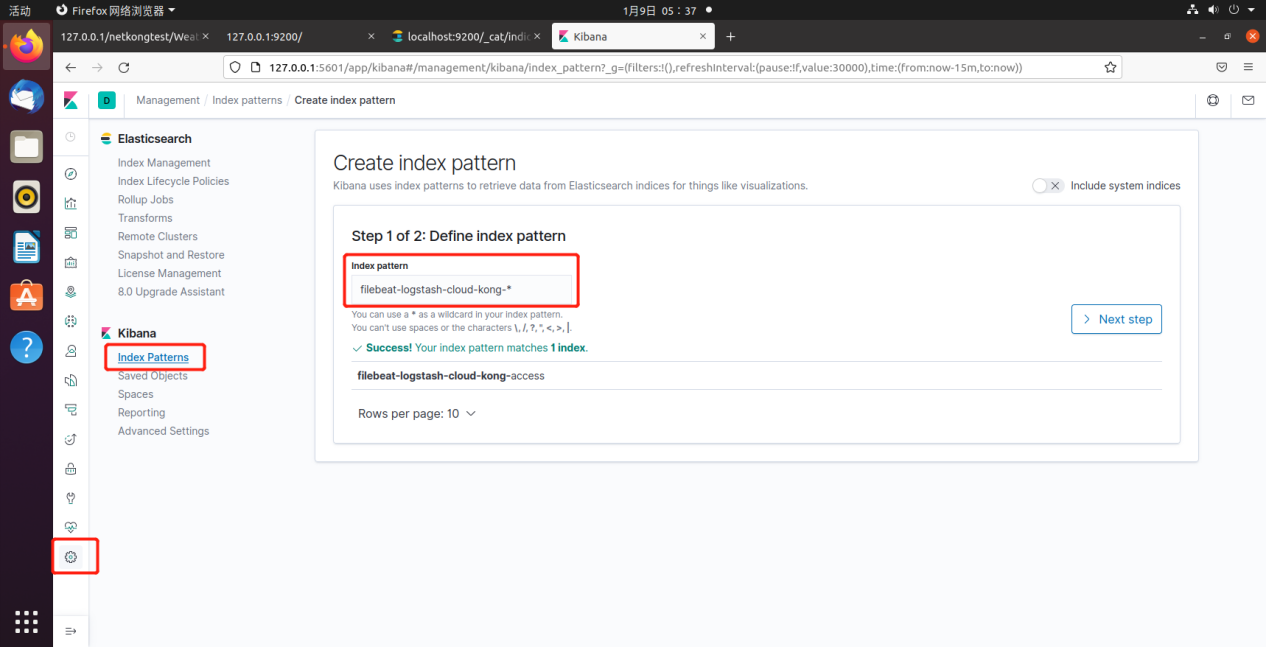
--restart=always \

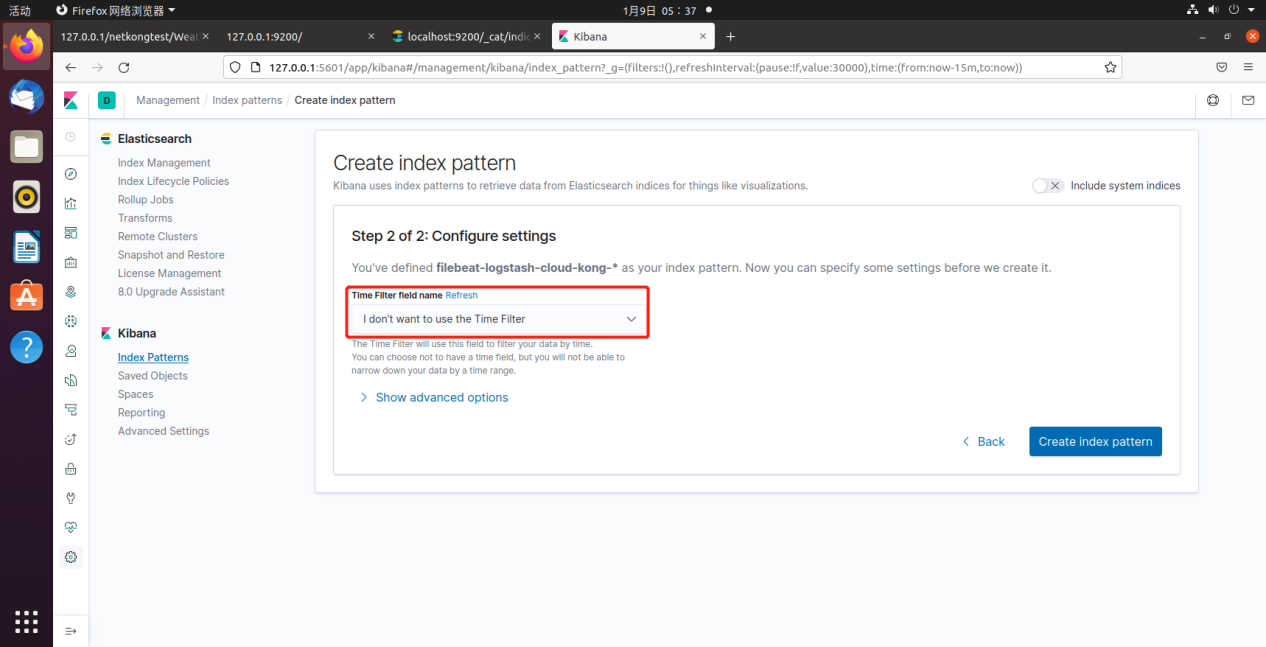
-p 5601:5601 \

-v /opt/kibana/config/kibana.yml:/usr/share/kibana/config/kibana.yml \

kibana:7.5.0

1. 测试检索功能
2. 访问127.0.0.1:5061端口，登录kibana
3. 创建一个index pattern





回到首页，我们可以添加过滤器，检索我们的日志了

