**一、需求**

使用Kafka做日志收集。

需要收集的信息：

1、用户ID（user\_id）

2、时间（act\_time）

3、操作（action，可以是：点击：click，收藏：job\_collect，投简历：cv\_send，上传简历：cv\_upload）

4、对方企业编码（job\_code）

**二、架构：**

HTML+Nginx+ngx\_kafka\_module+Kafka

**三、搭建过程**

1、安装配置zookeeper和kafka集群

  # 复用学习时的环境，不再赘述~

2、安装配置nginx

#下载Nginx

cd /opt/lagou/softwares

wget http://nginx.org/download/nginx-1.12.2.tar.gz

#解压nginx

tar zxvf nginx-1.18.0.tar.gz -C /opt/lagou/server

#安装ngx\_kafka\_module依赖librdkafka

yum install librdkafka-devel

#下载ngx\_kafka\_module

git clone https://github.com/brg-liuwei/ngx\_kafka\_module

#编译安装nginx

cd /opt/lagou/server/nginx-1.12.2

./configure --add-module=/opt/lagou/softwares/ngx\_kafka\_module

make && make install

#将可执行文件 nginx 软链接到 PATH 下：

ln -s /usr/local/nginx/sbin/nginx /usr/sbin/

#修改nginx配置文件：

vim /usr/local/nginx/conf/nginx.conf

#user  nobody;

worker\_processes  1;

#error\_log  logs/error.log;

#error\_log  logs/error.log  notice;

#error\_log  logs/error.log  info;

#pid        logs/nginx.pid;

events {

    worker\_connections  1024;

}

http {

    include       mime.types;

    default\_type  application/octet-stream;

    #log\_format  main  '$remote\_addr - $remote\_user [$time\_local] "$request" '

    #                  '$status $body\_bytes\_sent "$http\_referer" '

    #                  '"$http\_user\_agent" "$http\_x\_forwarded\_for"';

    #access\_log  logs/access.log  main;

    sendfile        on;

    #tcp\_nopush     on;

    #keepalive\_timeout  0;

    keepalive\_timeout  65;

    #gzip  on;

# 引入kafka模块

    kafka;

    # 指定broker地址

    kafka\_broker\_list linux121:9092 linux122:9092 linux123:9092;

    server {

        listen       80;

        server\_name  localhost;

        access\_log  logs/access.log;

        location / {

            root   html;

            index  index.html index.htm;

        }

        location = /homework/action {

                # 指定发送的 topic

         kafka\_topic topic\_homework;

        }

    }

    server {

        listen       80;

        server\_name  localhost;

        #charset koi8-r;

        #access\_log  logs/host.access.log  main;

        location / {

            root   html;

            index  index.html index.htm;

        }

        #error\_page  404              /404.html;

        # redirect server error pages to the static page /50x.html

        #

        error\_page   500 502 503 504  /50x.html;

        location = /50x.html {

            root   html;

        }

        # proxy the PHP scripts to Apache listening on 127.0.0.1:80

        #

        #location ~ \.php$ {

        #    proxy\_pass   http://127.0.0.1;

        #}

        # pass the PHP scripts to FastCGI server listening on 127.0.0.1:9000

        #

        #location ~ \.php$ {

        #    root           html;

        #    fastcgi\_pass   127.0.0.1:9000;

        #    fastcgi\_index  index.php;

        #    fastcgi\_param  SCRIPT\_FILENAME  /scripts$fastcgi\_script\_name;

        #    include        fastcgi\_params;

        #}

        # deny access to .htaccess files, if Apache's document root

        # concurs with nginx's one

        #

        #location ~ /\.ht {

        #    deny  all;

        #}

    }

    # another virtual host using mix of IP-, name-, and port-based configuration

    #

    #server {

    #    listen       8000;

    #    listen       somename:8080;

    #    server\_name  somename  alias  another.alias;

    #    location / {

    #        root   html;

    #        index  index.html index.htm;

    #    }

#}

    # HTTPS server

    #

    #server {

    #    listen       443 ssl;

    #    server\_name  localhost;

    #    ssl\_certificate      cert.pem;

    #    ssl\_certificate\_key  cert.key;

    #    ssl\_session\_cache    shared:SSL:1m;

    #    ssl\_session\_timeout  5m;

    #    ssl\_ciphers  HIGH:!aNULL:!MD5;

    #    ssl\_prefer\_server\_ciphers  on;

    #    location / {

    #        root   html;

    #        index  index.html index.htm;

    #    }

    #}

}

3、启动服务

#启动zookeeper

#通过之前的脚本群起

cd /root

sh zk.sh start

#后台启动kafka

kafka-server-start.sh -daemon /opt/lagou/servers/kafka\_2.12-1.0.2/config/server.properties"

#启动消费者消费主题：

kafka-console-consumer.sh --bootstrap-server linux43:9092 --topic topic\_homework

#启动nginx

nginx -s reload

4、编写html页面

#添加点击事件

<li ><button onclick="collect('click')">点击</button>

<li><button onclick="collect('collect')">收藏</button>

<li><button onclick="collect('send')">投简历</button>

<li><button onclick="collect('upload')">上传简历</button>

#点击事件js方法

 //收集操作日志

function collect(action) {

    let user\_id = ['user1', 'user2', 'user3']

    let job\_code = ['lagou', 'ali', 'tx'];

    $.ajax({

        url: 'http://linux121/homework/action',

        type: 'POST',

        contentType: 'application/json;charset=utf-8',

        dataType: 'json',

        data: {

            user\_id: user\_id[randomNum(0, 2)],

            act\_time: new Date().getTime(),

            action: action,

            job\_code: job\_code[randomNum(0, 2)]

        },

        success: function(data) {

        }

    })

}

//生成从minNum到maxNum的随机数

function randomNum(minNum, maxNum) {

    switch (arguments.length) {

        case 1:

            return parseInt(Math.random() \* minNum + 1, 10);

            break;

        case 2:

            return parseInt(Math.random() \* (maxNum - minNum + 1) + minNum, 10);

            break;

        default:

            return 0;

            break;

    }

}

# 上传 html 至 nginx /usr/local/nginx/html/homework目录下

**四、操作验证**

#访问linux121/homework/index.html

#点击操作，可见消费者打印日志

