

以图搜图 ubuntu配置java与nginx

2024年8月30日 14:53

1. 安装java:

```
sudo apt update
apt install openjdk-11-jre-headless
```

2. 安装nginx:

```
sudo apt-get install nginx
```

3. nginx配置:

3.1 服务器配置

\$ /etc/nginx: nginx配置文件目录。所有的nginx配置文件都在这里。

\$ /etc/nginx/nginx.conf: Nginx的主配置文件。可以修改他来改变nginx的全局配置。

\$ /etc/nginx/sites-available/: 这个目录存储每一个网站的"server blocks"。nginx通常不会使用这些配置, 除非它们陪连接到 sites-enabled 目录 (see below)。一般所有的server block 配置都在这个目录中设置, 然后软连接到别的目录。

\$ /etc/nginx/sites-enabled/: 这个目录存储生效的 "server blocks" 配置。通常,这个配置都是链接到 sites-available目录中的配置文件

\$ /etc/nginx/snippets: 这个目录主要可以包含在其它nginx配置文件中的配置片段。重复的配置都可以重构为配置片段。

3.2 日志文件

/var/log/nginx/access.log: 每一个访问请求都会记录在这个文件中, 除非你做了其它设置。

/var/log/nginx/error.log: 任何Nginx的错误信息都会记录到这个文件中。

3.3 修改配置文件: /etc/nginx/nginx.conf

```
user www-data;
worker_processes auto;
pid /run/nginx.pid;
include /etc/nginx/modules-enabled/*.conf;

events {
    worker_connections 768;
    # multi_accept on;
}

http {

    ##
    # Basic Settings
    ##

    sendfile on;
    tcp_nopush on;
    tcp_nodelay on;
    keepalive_timeout 65;
    types_hash_max_size 2048;
    # server_tokens off;

    # server_names_hash_bucket_size 64;
    # server_name_in_redirect off;

    include /etc/nginx/mime.types;
    default_type application/octet-stream;

    ##
    # SSL Settings
    ##

    # ssl_protocols TLSv1 TLSv1.1 TLSv1.2 TLSv1.3; # Dropping SSLv3, ref:
```

```

POODLE
# ssl_prefer_server_ciphers on;

##
# Logging Settings
##

access_log /var/log/nginx/access.log;
error_log /var/log/nginx/error.log;

##
# Gzip Settings
##

gzip on;

# gzip_vary on;
# gzip_proxied any;
# gzip_comp_level 6;
# gzip_buffers 16 8k;
# gzip_http_version 1.1;
# gzip_types text/plain text/css application/json application/javascript
text/xml application/xml application/xml+rss text/javascript;

##
# Virtual Host Configs
##

# include /etc/nginx/conf.d/*.conf;
# include /etc/nginx/sites-enabled/*;

server {
    listen    8080;
    server_name localhost;

    add_header Access-Control-Allow-Origin *;
    add_header Access-Control-Allow-Methods 'GET,POST';
    add_header Access-Control-Allow-Headers 'DNT,X-Mx-
ReqToken,Keep-Alive,User-Agent,X-Requested-With,If-Modified-
Since,Cache-Control,Content-Type,Authorization';

    #charset koi8-r;

    #access_log logs/host.access.log main;

    location / {
        root /home/image_portal/dist;
        index index.html index.htm;
    }
}

```

```
#####
```

```

server {
    listen    8080;
    server_name localhost;

    add_header Access-Control-Allow-Origin *;
    add_header Access-Control-Allow-Methods 'GET,POST';
    add_header Access-Control-Allow-Headers 'DNT,X-Mx-ReqToken,Keep-
Alive,User-Agent,X-Requested-With,If-Modified-Since,Cache-
Control,Content-Type,Authorization';

    #charset koi8-r;

    #access_log logs/host.access.log main;

```

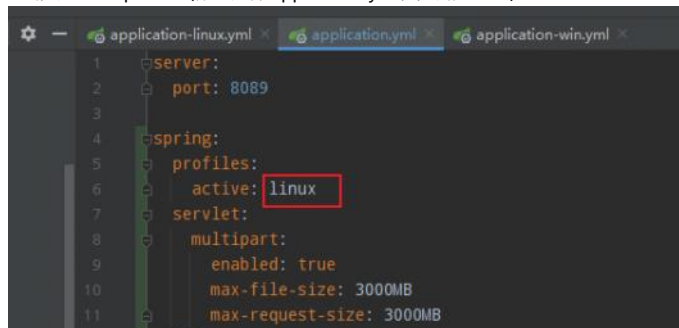
```

location / {
    root /home/image_portal/dist;
    index index.html index.htm;
}
}
#####

```

4. 修改java 后端配置文件

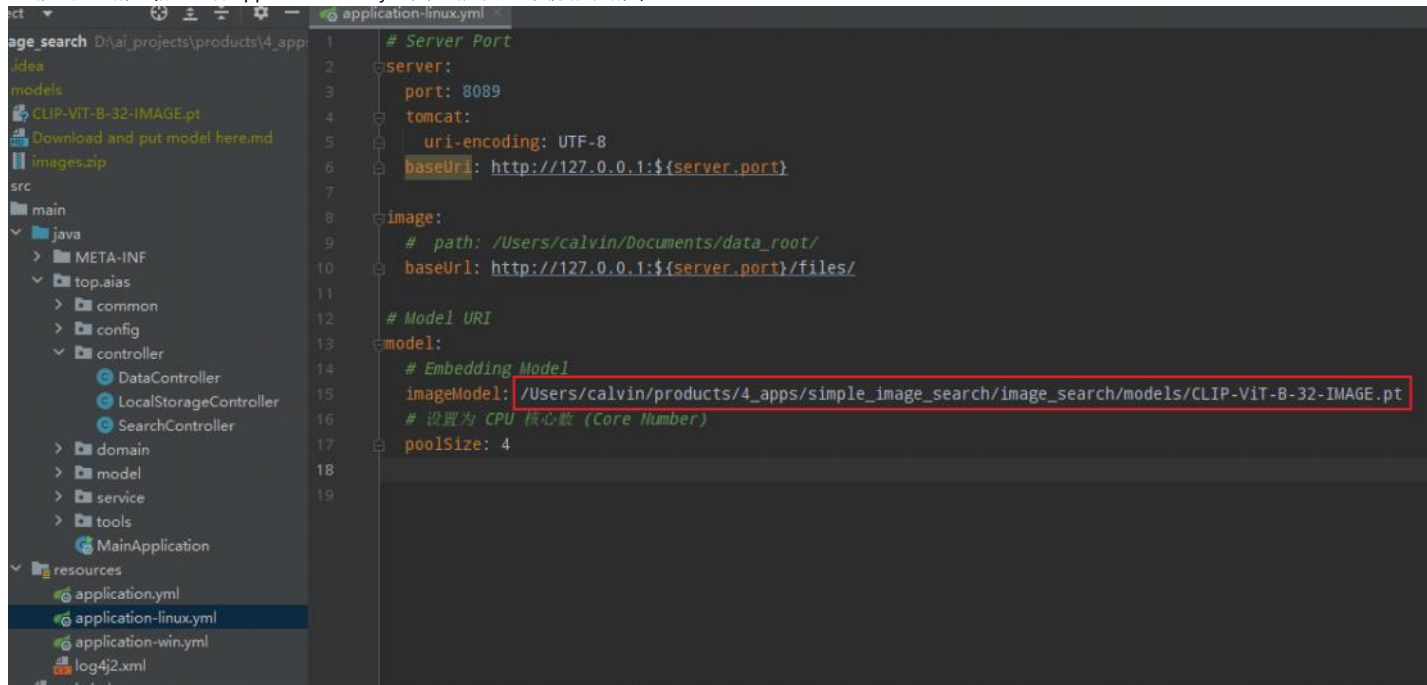
4.1 修改active profile (配置文件 application.yml, 更新为 linux)



4.2 修改ip地址 (配置文件 application-linux.yml, 更新为主机ip地址)



4.3 修改模型路径 (配置文件 application-linux.yml, 更新为模型的实际存放路径)

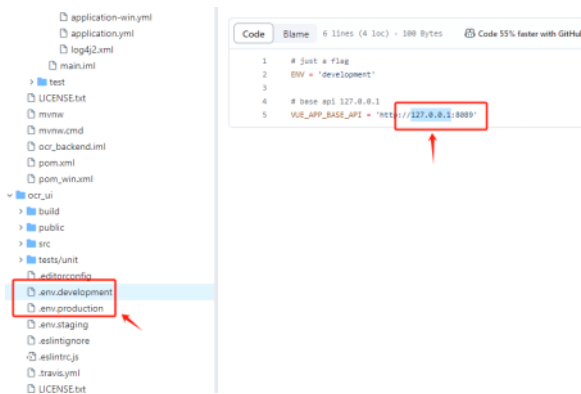


5. 运行 java

nohup java -Dfile.encoding=utf-8 -jar image_search-0.23.0.jar > log.txt 2>&1 &

6. 前端配置运行

6.1 前端配置: 更新后端的ip地址



6.2 运行前端程序

安装node modules:

```
npm install
```

运行前端程序:

```
npm run dev
```

构建部署包:

```
npm run build:prod
```

7. nginx 启动

```
sudo systemctl reload nginx
```

```
sudo systemctl restart nginx
```

```
#####
```

Nginx 常用命令:

停止nginx:

```
sudo systemctl stop nginx
```

启动nginx:

```
sudo systemctl start nginx
```

重启nginx:

```
sudo systemctl restart nginx
```

修改配置文件后,平滑加载配置命令(不会断开用户访问):

```
sudo systemctl reload nginx
```

默认,nginx是随着系统启动的时候自动运行。如果你不想开机启动,那么你可以禁止

nginx开机启动:

```
sudo systemctl disable nginx
```

重新配置nginx开机自动启动:

```
sudo systemctl enable nginx
```

```
#####
```

8. 运行测试:

<http://xxx.xx.xx.xxx:8080/>