项目部署说明文档

# 简述

本项目部署在Ubuntu20.04服务器上，绑定域名komeij.cyou，并签发了SSL证书，可以通过<https://komeiji.cyou>访问该项目的主页进行使用

本项目是基于SpringBoot+Vue3的前后端分离Web程序，代码托管在<https://github.com/KomeijiHelper/KomeijiHelper>仓库中，前后端程序均打包为可部署产物上传到服务器中

前端的vue程序打包得到名为dist的文件夹，然后通过nginx提供静态文件服务，将其代理到80以及443端口，提供http和https服务，并通过axios发起API请求，以nginx代理到后端提供的服务

# 系统以及依赖

操作系统：Ubuntu20.04

Glibc：2.31

Java：openjdk 17.0.14

Nodejs：22.14.0

MySQL：8.0.41

Redis：5.0.7

Nginx：1.18.0

Certbot：0.40.0

# 部署流程

1. 获取代码，分别编译并打包前后端得到前端产物dist文件夹以及后端产物komeiji-0.0.1-SNAPSHOT.jar
2. 整个项目位于/root/komeiji/ 目录下，将前后端打包产物放于此处
3. 签发SSL证书，使用certbot自动配置SSL并生成证书
4. 配置nginx，在/etc/nginx下的默认配置上

首先修改nginx.conf中的user为root，以防出现权限问题

再于/etc/nginx/site-enabled里面添加配置文件，具体内容如下

1. server {
2. listen 80;
3. server\_name komeiji.cyou;
5. location / {
6. **return** 301 https://$host$request\_uri;
7. }
8. }
10. server {
11. listen 443 ssl;
12. server\_name komeiji.cyou;
14. client\_max\_body\_size 50m;
16. ssl\_certificate /etc/letsencrypt/live/komeiji.cyou/fullchain.pem;
17. ssl\_certificate\_key /etc/letsencrypt/live/komeiji.cyou/privkey.pem;
19. ssl\_protocols TLSv1.2 TLSv1.3;
20. ssl\_ciphers HIGH:!aNULL:!MD5;
22. location / {
23. root /root/komeiji/dist;
24. index index.html index.htm;
25. try\_files $uri $uri/ /index.html;
26. }
28. error\_page 404 /404.html;
29. location = /404.html {
30. root /usr/share/nginx/html;
31. }
33. error\_page 500 502 503 504 /50x.html;
34. location = /50x.html {
35. root /usr/share/nginx/html;
36. }
38. location /api/ {
39. proxy\_pass http://localhost:8081/;
40. proxy\_pass\_request\_headers on;
41. proxy\_set\_header Host $host;
42. proxy\_set\_header X-Real-IP $remote\_addr;
43. proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;
44. proxy\_set\_header Cookie $http\_cookie;
45. proxy\_set\_header Authorization $http\_authorization;
46. }
48. location /file/image/ {
49. alias /var/www/file/image/;
50. autoindex on;
51. }
53. }

在38行中我们设置了前端通过komeji.cyou/api/前缀来访问运行于服务器8081端口的后端服务；因为本项目支持发送图片，我们将所有的图片压缩并hash重命名于/var/www/file/image目录下，以/file/image前缀来提供图床服务

1. 开启对应的mysql和redis服务，推荐使用systemd挂载服务，mysql初始化并创建数据库komeiji
2. 通过` java -jar /root/komeiji/komeiji-0.0.1-SNAPSHOT.jar --spring.profiles.active=prod` 命令运行后端程序，此处使用application-prod.yaml的生产环境配置，需要注意根据redis和mysql的密码修改对应的配置