

项目部署

nginx服务器

- 安装: `sudo apt install nginx`
- 说明: 简单使用nginx
- 添加虚拟主机:
 - 在 `/etc/nginx/sites-available` 下创建虚拟主机配置文件 `www.bbs.com` , 内容如下:

```
server {  
    listen 80;  
    server_name www.bbs.com bbs.com;  
  
    location / {  
        root /var/www/blog;  
        index index.html;  
    }  
}
```

- 创建软链接: 在sites-enabled下创建

```
sudo ln -s /etc/nginx/sites-available/www.bbs.com  
/etc/nginx/sites-enabled/www.bbs.com
```

- 重启nginx服务
 - 创建站点目录: `/var/www/bbs`

Flask项目

- 说明: 将flask项目拷贝到 `/var/www/bbs` 目录下

uWSGI

- 说明：uWSGI是实现了WSGI协议的应用程序。
- 安装：`pip3 install uwsgi`
- 配置：

```
http           # 以http协议启动
socket         # 以socket方式启动
chdir          # uwsgi启动的项目的根目录
wsgi-file      # 数据交给哪个模块处理
callable       # 具体可调用的对象
daemonize      # 后台运行(需要指定一个日志文件)
processes      # 指定使用的进程数
threads       # 指定使用的线程数
```

- 启动演示(以http协议启动)

```
uwsgi --http 127.0.0.1:5000 --wsgi-file manage.py --callable app
```

- socket方式启动
 - 需要nginx转发请求

```
server {
    listen 80;
    server_name www.bbs.com bbs.com;

    location / {
        # 包含请求参数
        include uwsgi_params;
        # 以socket方式转发请求
        uwsgi_pass 127.0.0.1:5000;
    }
}
```

一定不要忘记重新启动nginx。

- 启动uwsgi

```
uwsgi --socket 127.0.0.1:5000 --wsgi-file manage.py --callable
app
```

- 将启动参数写入配置文件 `uwsgi.ini`

```
[uwsgi]
socket = 127.0.0.1:5000
wsgi-file = manage.py
callable = app
# 后台启动
daemonize = ~/log/uwsgi.log
```

启动: uwsgi uwsgi.ini

静态资源处理

- 说明: 静态文件的请求没有必要转发, 因为nginx自己就可以处理。所以我们想要重写转发规则。
- 静态资源的路由转发规则:

```
location /static {
    root /var/www/bbs;
    # 或
    alias /var/www/bbs/static;
}
```

- 测试准备
 - 在项目目录下创建 `static` 文件夹
 - 拷贝一张图片 `gyy.jpeg` 到 `static` 目录
- 测试: 看到图片表示OK

`www.bbs.com/static/gyy.jpeg`

上线步骤:

1. 在线上服务器安装虚拟开发环境
2. 将开发环境中依赖包 `require.txt` 中的所有依赖在虚拟开发环境中安装一遍

```
pip install -r require.txt
```

3. 在虚拟开发环境中安装uwsgi

```
pip install uwsgi
```

4. 将自己的项目拷贝到站点根目录

- 创建配置文件,将启动参数写入配置文件 `uwsgi.ini`

```
[uwsgi]
socket = 127.0.0.1:5000
wsgi-file = manage.py
callable = app
# 后台启动
daemonize = ~/log/uwsgi.log
```

- 启动uwsgi (在虚拟开发环境中)

```
uwsgi uwsgi.ini
```

```
ps -aux|grep uwsgi  查看是否启动
```

5. 配置虚拟站点

```
cd /etc/nginx/sites-available/
#以default为模板创建配置文件
sudo cp default www.1904.com
```

```
server {
    listen 80;
    root /data/www/1904;
    server_name www.1904.com;

    location / {
        # 配置uwsgi
        include uwsgi_params;
        uwsgi_pass 127.0.0.1:5050;
    }
    # 静态文件请求处理
    location /static {
        alias /data/www/1904/static;
    }
}
```

```
cd /etc/nginx/sites-enable/  
sudo ln -s ../sites-available/www.1904.com www.1904.com  
#重启nginx  
sudo service nginx restart
```

6 把项目中使用数据库备份出来,拷贝到服务器上,在服务器重建数据库,然后把数据恢复到数据库中