**Django + Uwsgi + Nginx+UbuntuServer的生产环境部署**

使用runserver可以使我们的django项目很便捷的在本地运行起来，但这只能在局域网内访问，如果在生产环境部署django，就要多考虑一些问题了。比如静态文件处理，安全，效率等等，总结归纳基于uwsgi+Nginx下django项目生产环境的部署

准备条件

pip install uwsgi

1.使用命令启动uwsgi

第一步：进入django项目

第二步：命令测试启动

uwsgi --http IP:HOST --file PROJECT/wsgi.py --static-map=/static=static

参数说明：

--http 这个就和runserver一样指定IP 端口

--file 这个文件就里有一个反射,如果你在调用他的时候没有指定Web Server就使用默认的

-- static 做一个映射,指定静态文件

1. 使用uwsgi配置文件启动django项目

第一步：在django项目同级目录创建script目录，用于存放配置脚本等等

第二步：进入/script目录，创建一个uwsgi.ini文件

编辑uwsgi.ini文件内容如下：

# uwsig使用配置文件启动

[uwsgi]

# 项目目录

chdir=/opt/proj/PROJECT/

# 指定项目的application

module=teacher.wsgi:application

# 指定sock的文件路径

socket=/opt/proj/script/uwsgi.sock

# 进程个数

workers=5

pidfile=/opt/proj/script/uwsgi.pid

# 指定IP端口

http=IP：HOST

# 指定静态文件

static-map=/static=/opt/proj/PROJECT/static

# 启动uwsgi的用户名和用户组

uid=root

gid=root

# 启用主进程

master=true

# 自动移除unix Socket和pid文件当服务停止的时候

vacuum=true

# 序列化接受的内容，如果可能的话

thunder-lock=true

# 启用线程

enable-threads=true

# 设置自中断时间

harakiri=30

# 设置缓冲

post-buffering=4096

# 设置日志目录

daemonize=/opt/proj/script/uwsgi.log

启动项目：

uwsgi --ini uwsgi.ini

至此，uwsgi+django就完美结合了，但是，光有uwsgi还不够，uwsgi处理动态请求能力高，但对于静态请求（如static文件，css,js文件等）处理能力差，此时就要结合nginx一起使用

**Uwsgi + Nginx**

安装nginx

sudo apt-get install nginx

安装好的文件位置：

/usr/sbin/nginx：主程序

/etc/nginx：存放配置文件

/usr/share/nginx：存放静态文件

/var/log/nginx：存放日志

启动：nginx start

停止：nginx stop

重启：nginx restart

配置文件

首先创建配置文件，nginx的默认配置目录为/etc/nginx/conf.d

cd /etc/nginx/conf.d/

创建一个配置文件：PROJECT.conf

server { # 这个server标识我要配置了

listen 80; # 我要监听那个端口

server\_name 192.168.2.108 ; # 你访问的路径前面的url名称

access\_log /var/log/nginx/access.log main; # Nginx日志配置

charset utf-8; # Nginx编码

gzip\_types text/plain application/x-javascript text/css text/javascript application/x-httpd-php application/json text/json image/jpeg image/gif image/png application/octet-stream; # 支持压缩的类型

error\_page 404 /404.html; # 错误页面

error\_page 500 502 503 504 /50x.html; # 错误页面

# 指定项目路径uwsgi

location / { # 如同Django的url(r'^admin/', admin.site.urls),

include uwsgi\_params; # 导入一个Nginx模块他是用来和uWSGI进行通讯的

uwsgi\_connect\_timeout 30; # 设置连接uWSGI超时时间

uwsgi\_pass unix:/opt/proj/script/uwsgi.sock; # 指定uwsgi的sock文件所有动态请求就会直接丢给他

}

# 指定静态文件路径

location /static/ {

alias /opt/proj/PEOJECT/static/;

index index.html index.htm;

}

}

重启Nginx