



uwsgi1

一：测试uwsgi

1. pip install uwsgi
2. apt-get install build-essential python-dev  依赖
3. 新建个test.py文件

def application(env, start\_response):

    start\_response('200 OK', [('Content-Type','text/html')])

    return [b"Hello World"]

1. 查看uwsgi开启状态 netstat -tulnp ps -ef | grep uwsgi
2. 查看防火墙(firewall)服务状态 systemctl status firewalld

查询端口是否开放 firewall-cmd --query-port=9090/tcp

开放9090端口 firewall-cmd --permanent --add-port=9090/tcp

重启防火墙(修改配置后要重启防火墙） firewall-cmd --reload

1. 运行uwsgi

uwsgi --http :9090 --wsgi-file test.py

1. 查看uwsgi开启状态 netstat -tulnp 以及ps -ef | grep uwsgi
2. 测试uwsgi是否可成功运行:

访问ip:9090 页面输出"Hello World"则表示成功

二：为你的项目创建 your\_uwsgi.ini

[uwsgi]

**chdir** = /wodewangzhan/mysite //指定项目运行根目录

module = mysite.wsgi:application //指定使用你根目录下的website里面的wsgi

**socket** = 127.0.0.1:8000 //监听到网络套接字

master = true //主进程启动

**virtualen**v = /wodewangzhan/for\_mysite //指定你虚拟环境的位置

#vhost = true //多站模式

#no-site = true //多站模式时不设置入口模块和文件

#workers = 2 //子进程数

#reload-mercy = 10

#vacuum = true //退出、重启时清理文件

#max-requests = 1000

#limit-as = 512

#buffer-size = 30000

#pidfile = /var/run/uwsgi9090.pid //pid文件，用于下脚本启动、停止该进程

daemonize = /wodewangzhan/mysite/run.log //日志

disable-logging = true

uwsgi命令：

1. uwsgi --ini you\_uwsgi.ini //启动uwsgi,uwsgi已经添加到bin下 //如 uwsgi: /usr/bin/uwsgi
2. killall -9 uswgi //杀死全部uwsgi

nginx:

nginx命令：

1. /usr/local/nginx/sbin/nginx //启动nginx
2. 如果端口被占用需要killall -9 nginx或者查看防火墙是否开放80端口
3. /usr/local/nginx/sbin/nginx -t //查看nginx默认配置路径和配置情况

为你的项目编辑nginx配置

server {

**listen 80; //监听80端口**

**server\_name xxxx.com; //你的域名**

client\_max\_body\_size 500M; //允许上传最大的文件为500M默认1M

location / {

#root html;

#index index.html index.htm;

**include uwsgi\_params;**

**uwsgi\_pass 127.0.0.1:8000; //与你your\_uwsgi.ini socket一致**

}

**location /static** { //你的静态文件位置

alias /wodewangzhan/mysite/all\_static\_files;

}

**location /media** { //你的图像等位置

alias /wodewangzhan/mysite/media;

}

}

注意多查看以下内容：

1. 要查看防火墙是否开放了nginx和uwsgi等你需要服务的端口
2. ps -ef | grep nginx 或者 netstat -tulnp 查看端口访问配置
3. 修改服务后要重启服务

Django

1. 在wodewangzhan下创建虚拟环境for\_mysite

virtualenv –p 你本地的python解释器 --no-site-packages for\_mysite

1. pull你远方的代码或者上传本地代码
2. pip install -r requirements.txt 批量安装所需库
3. 然后修改DEBUG=False, ALLOWED\_HOSTS = ["ip",".域名"].表示可以访问二级域名
4. STATIC\_ROOT = os.path.join(BASE\_DIR, “all\_static\_files”)即在wodewangzhan/mysite下
5. Python manage.py collectstatic --noinput 收集静态文件
6. 生产数据表迁移以及创建表,然后在manage.py 同级目录下执行your\_uwsgi.ini
7. .....

在Pycharm中通过SFTP把远端的项目移到本地

1.点击tools

2.点击Deploment下的Configuration

3.输入主机位置，密码等

4.点击Mappings(映射)，选择映射远端文件位置

5.点击Deploment下的Download to ...把远方文件下载下来

6.本地编辑，远方自动更新，点击Deploment下的Automatic Upload always

7.看远方服务器文件 点击Deploment下的browse remote hos

8. 配置远方虚拟环境的解释器到本地