**Django Nginx+uwsgi 安装配置**

在前面的章节中我们使用 **python manage.py runserver** 来运行服务器。这只适用测试环境中使用。

正式发布的服务，我们需要一个可以稳定而持续的服务器，比如apache, Nginx, lighttpd等，本文将以 Nginx 为例。

**安装基础开发包**

Centos 下安装步骤如下：

yum groupinstall "Development tools"

yum install zlib-devel bzip2-devel pcre-devel openssl-devel ncurses-devel sqlite-devel readline-devel tk-devel

CentOS 自带 Python 2.4.3，但我们可以再安装Python2.7.5：

cd ~

wget http://python.org/ftp/python/2.7.5/Python-2.7.5.tar.bz2

tar xvf Python-2.7.5.tar.bz2

cd Python-2.7.5

./configure --prefix=/usr/local

make && make altinstall

**安装Python包管理**

easy\_install包 <https://pypi.python.org/pypi/distribute>

安装步骤:

cd ~

wget https://pypi.python.org/packages/source/d/distribute/distribute-0.6.49.tar.gz

tar xf distribute-0.6.49.tar.gz

cd distribute-0.6.49

python2.7 setup.py install

easy\_install --version

pip包: <https://pypi.python.org/pypi/pip>

安装pip的好处是可以 pip list、pip uninstall 管理Python包， easy\_install 没有这个功能，只有uninstall

**安装uwsgi**

uwsgi:<https://pypi.python.org/pypi/uWSGI>

uwsgi参数详解：<http://uwsgi-docs.readthedocs.org/en/latest/Options.html>

pip install uwsgi

uwsgi --version #查看 uwsgi 版本

测试uwsgi是否正常：

新建test.py文件，内容如下：

def application(env, start\_response):

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

return "Hello World"

然后在终端运行：

uwsgi --http :8001 --wsgi-file test.py

在浏览器内输入：http://127.0.0.1:8001，查看是否有"Hello World"输出，若没有输出，请检查你的安装过程。

**安装 Django**

pip install django

测试django是否正常，运行：

django-admin.py startproject demosite

cd demosite

python2.7 manage.py runserver 0.0.0.0:8002

在浏览器内输入：http://127.0.0.1:8002，检查django是否运行正常。

**安装 Nginx**

安装命令如下：

cd ~

wget http://nginx.org/download/nginx-1.5.6.tar.gz

tar xf nginx-1.5.6.tar.gz

cd nginx-1.5.6

./configure --prefix=/usr/local/nginx-1.5.6 \

--with-http\_stub\_status\_module \

--with-http\_gzip\_static\_module

make && make install

你可以阅读 [Nginx 安装配置](http://www.w3cschool.cc/linux/nginx-install-setup.html) 了解更多内容。

**uwsgi 配置**

uwsgi支持ini、xml等多种配置方式，本文以 ini 为例， 在/ect/目录下新建uwsgi9090.ini，添加如下配置：

[uwsgi]

socket = 127.0.0.1:9090

master = true //主进程

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，用于下面的脚本启动、停止该进程

daemonize = /website/uwsgi9090.log

**Nginx 配置**

找到nginx的安装目录（如：/usr/local/nginx/），打开conf/nginx.conf文件，修改server配置：

server {

listen 80;

server\_name localhost;

location / {

include uwsgi\_params;

uwsgi\_pass 127.0.0.1:9090; //必须和uwsgi中的设置一致

uwsgi\_param UWSGI\_SCRIPT demosite.wsgi; //入口文件，即wsgi.py相对于项目根目录的位置，“.”相当于一层目录

uwsgi\_param UWSGI\_CHDIR /demosite; //项目根目录

index index.html index.htm;

client\_max\_body\_size 35m;

}

}

你可以阅读 [Nginx 安装配置](http://www.w3cschool.cc/linux/nginx-install-setup.html) 了解更多内容。

设置完成后，在终端运行：

uwsgi --ini /etc/uwsgi9090.ini &

/usr/local/nginx/sbin/nginx

在浏览器输入：http://127.0.0.1，你就可以看到django的"It work"了。