

(/apps/redi
utm_sourc
banner-clic

将Django项目部署到服务器



一许青衫一 (/u/8c6979d8ab4f) +关注

2018.05.22 00:02* 字数 2355 阅读 715 评论 0 喜欢 14

(/u/8c6979d8ab4f)

一、概述

在本地搭建好了个人博客之后，接着需要将其部署到服务器上面。在此之前，我们需要做一些准备工作。

1.1 准备工具

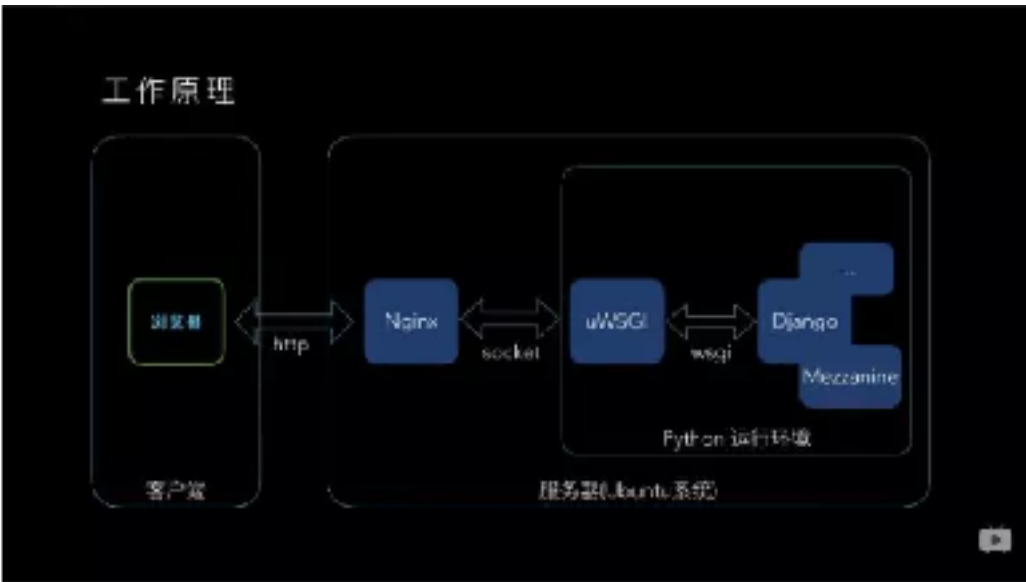
部署网站到服务器之前，我们需要拥有一个云主机与域名，这两者都需要购买。目前已经拥有。

1.2 相关基础概念

在部署网站到服务器上面的过程中，涉及到一些新的东西，要理解其作用。参考文章《web建站过程中流程》。现在基本知道了部署网站所需要的各项东西，这个个人博客的架构是Linux + Nginx+MySQL+Python，Framework是Django。

(https://ds
click.youd
slot=30edc
db4f-4715-
9dce3945e

二、流程图解



流程图

具体流程：

当客户端访问服务器的时候，其实是浏览器和服务器上的 Web Server（Web服务）产生了一个HTTP协议的通信。上图Python编写的源代码是运行在Python运行环境中的，并且是封装于Python的虚拟环境venv中。



接下来用到了uWSGI模块，这是属于Python的模块，（**此模块在Win上运行错误，在Linux上运行良好**）。uWSGI一方面通过WSGI与Python代码进行通信，另一方面通过socket（套接字）与Nginx通信。

三、WSGI

3.1 WSGI 是什么？

WSGI 的全称是*Web Server Gateway Interface* ,翻译过来就是Web服务器网关接口。具体来说，

WSGI是一个规范，定义了Web服务器如何与Python程序进行交互，使得使用Python写的Web应用程序可以和Web服务器对接起来。

3.2 为什么需要WSGI这个规范？

在Web部署的方案上，有一个方案是目前应用最广泛的：

首先，部署一个Web服务器专门用来处理HTTP协议层面相关的事情，比如如何在一个物理机上提供多个不同的Web服务(单IP多域名，单IP多端口等)这种事情。

然后，部署一个用各种语言编写（Java，PHP，Python，Ruby）的应用程序，这个应用程序会从Web服务器上接收客户端的请求，处理完成后，再返回响应给Web服务器，最后由Web服务器返回给客户端。

那么，要采用这种方案，Web服务器和应用程序之间就要知道如何进行交互。为了定义Web服务器和应用程序之间的交互过程，就形成了很多不同的规范。如Java专用的Servlet规范，还有Python专用的WSGI规范等。提出这些规范的目的就是1为了定义统一的标准，提升程序的可移植性。

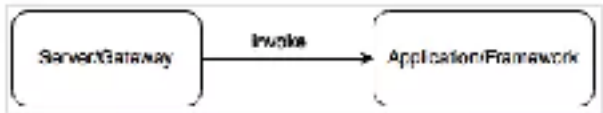
3.3 WSGI 是怎么工作的

WSGI 相当于是Web服务器和Python应用程序之间的桥梁。其存在的目的有两个：

- 1、让Web服务器知道如何调用Python应用程序，并且把用户的请求告诉应用程序。
- 2、让Python应用程序知道用户的具体请求是什么，以及如何返回结果给Web服务器。

3.4 WSGI中的角色

在WSGI中定义了两个角色，Web服务器端称为server或者gateway，应用程序端称为application或者framework（因为WSGI的应用程序端的规范一半都是由具体的框架来实现的）。



WSGI中的角色



3.5 server 如何调用application

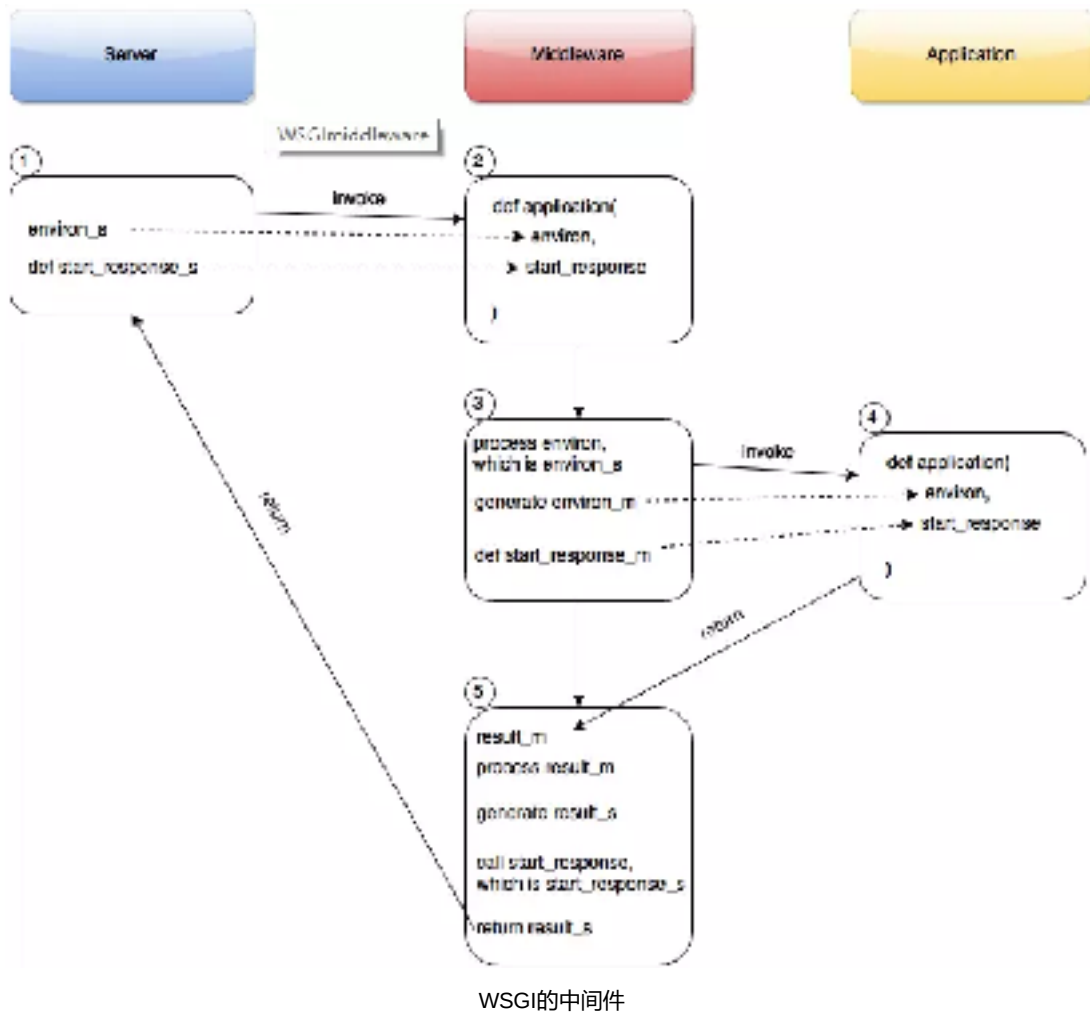
首先，每个application的入口只有一个，也就是所有的客户端请求都同一个入口进入到应用程序。

(/apps/redi
utm_sourc
banner-cli

接下来，server端需要知道去哪里找application的入口。这个需要在server端指定一个Python模块，也就是Python应用中的一个文件，并且这个模块中需要包含一个名称为application的可调用对象（函数和类都可以），这个application对象就是这个应用程序的唯一入口。

3.6、WSGI 中间件

WSGI Middleware（中间件）也是WSGI规范的一部分。现在已经知道了WSGI中的两个角色：server与application。那么middleware就是一种运行在server和application中的应用（一般都是Python应用）。middleware同时具备server和application角色，对于server来说，他是一个application，而对于application来说，他是一个server。middleware并不修改server端和application端的规范，只是同时实现了这两个角色的功能而已。



middleware工作流程如下：

1、Server收到客户端的HTTP请求后，生成了 environ_s，并且已经定义了 start_response_s。



2、Server调用Middleware的application对象，传递的参数是 environ_s 和 start——response_s。

3、Middleware 回根据 environ 执行业务逻辑，生成 environ_m，并且已经定义了 start_response_m。

(/apps/redi
utm_sourc
banner-clic

4、Middleware 决定调用Application的application对象，传递参数是 environ_m 和 start_response_m。

5、Middleware处理result_m，然后生成 result_s,接着调用 start_response_s，并返回结果 result_s给Server端。Server端获取到result_s后就可以发送结果给客户端了。

从上面的流程可以看出middleware应用的几个特点：

1、Server认为middleware是一个application。

2、Application认为middleware是一个server。

3、Middleware可以有多层。

因为Middleware能处理所有经过的request和response，所有要做什么都可以，没有限制。比如可以检测request是否有非法内容，检查response是否有非法内容，为request加上特定的HTTP header等，这些都是可以的。

3.7、WSGI的实现和部署

要使用WSGI，需要分别实现server和application角色。

Application端的实现一般是由Python的各种框架来实现的，比如Django，web.py等，一般开发者不需要关心WSGI的实现，框架会提供接口让开发者获取HTTP请求的内容以及发送HTTP响应。

Server端的实现会比较复杂一点，这个主要是因为软件架构的原因。一般常用的Web服务器，如Apache和Nginx，都不会内置WSGI的支持，而是通过扩展来完成。比如Apache服务器，会通过扩展模块mod_wsgi来支持WSGI。Apache和mod_wsgi之间通过程序内部接口传递信息，mod_wsgi会实现WSGI的server端、进程管理以及对application的调用。Nginx上一般是用proxy的方式，用nginx的协议将请求封装好，发送给应用服务器，比如uWSGI，应用服务器会实现WSGI的服务端、进程管理以及对application的调用。

参考信息：<https://segmentfault.com/a/1190000003069785>

四、ubuntu

目前个人博客的大致框架已经代码敲好了，选用 Nginx+uwsgi 部署到 center os 系统的服务器上面。但再找资料进行部署时，关于uwsgi模块的安装，在win上失败。谷歌一下，发现，uwsgi模块不支持win，但支持ubuntu。所以目前情况变为把代码迁移到ubuntu系统上运行，然后在部署到服务器上面。



4.1 ubuntu上部署流程

1、安装Nginx

```
apt-get install nginx
```

安装Nginx

(/apps/redi
utm_sourc
banner-cl

ubuntu安装完Nginx之后，文件结构大致为：

所有的配置文件都在/etc/nginx下；启动程序文件在/usr/sbin/nginx下；日志文件在/var/log/nginx/下，分别是access.log和error.log；

并且在/etc/init.d下创建了启动脚本nginx。

```
sudo /etc/init.d/nginx start
sudo /etc/init.d/nginx stop
sudo /etc/init.d/nginx restart
```

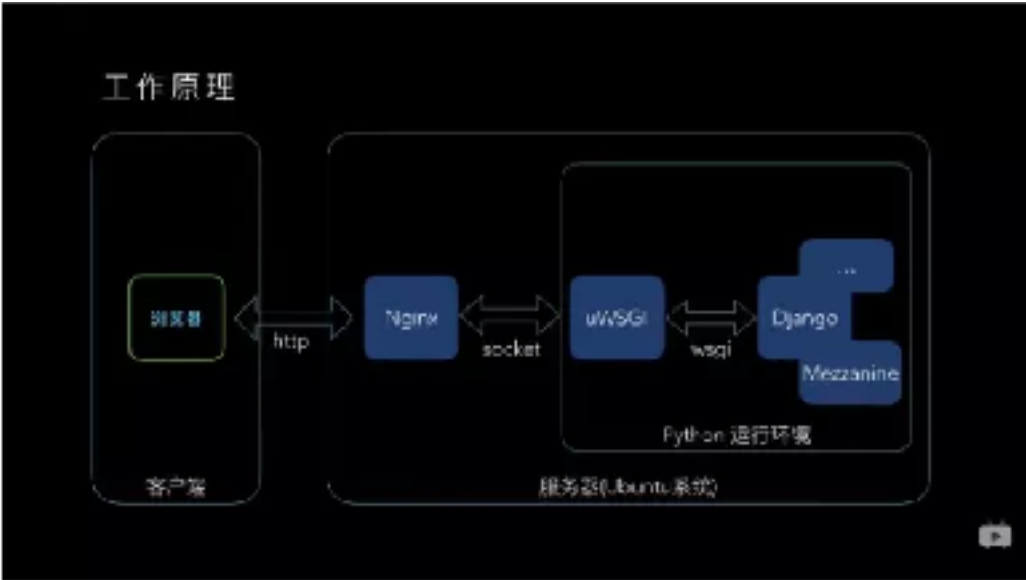
操作Nginx命令

2、安装uwsgi

```
apt-get install python-dev
```

```
pip install uwsgi
```

这里大体的流程是：Nginx作为服务器的最前端，负责接收client的所有请求，统一管理。静态请求由Nginx自己处理。非静态请求通过uwsgi传递给Django，由Django来进行处理，从而完成一次Web请求。



流程图

3、测试uwsgi

在Django项目下新建test.py文件，



```
# test.py
def application(environ, start_response):
    status, response_headers = headers_from_dict({'Content-Type': 'text/plain'})
    return status, response_headers, ['Hello World!']
```

测试uwsgi

(/apps/redi
utm_sourc
banner-cli

然后执行shell命令：

```
uwsgi --http :8001 --plugin python --wsgi-file test.py
```

加上--plugin python 是告诉uWSGI在使用python插件，不然很有可能会出现类似这样的错误：

```
uwsgi: unrecognized option '--wsgi-file'
getopt._load() error
```

错误

执行成功在浏览器中打开：http://localhost:8001 显示 Hello World 说明uwsgi正常运行。

4、测试Django

首先得保证Django项目没有问题

```
python manage.py runserver 0.0.0.0:8001
```

访问http://localhost:8001,项目运行正常。

然后链接Django和uwsgi，实现简单的web服务器，到Django项目目录下执行shell：

```
uwsgi --http :8001 --plugin python --module blog.wsgi
```

blog为你的项目名。访问http://localhost:8001, 项目正常。注意这时项目的静态文件是不会被加载的，需要用Nginx做静态文件代理。

5、配置 uwsgi

uwsgi支持通过配置文件的方式启动，可以接受更多的参数，高度可定制。我们在Django项目目录下新建uwsgi.ini



```

# Django-related settings
SECRET_KEY = '...'

# the base directory (full path)
MEDIA_ROOT = '/home/ubuntu/...'

```

(/apps/redi
utm_sourc
banner-clic

配置uwsgi

6、配置Nginx

```

# nginx默认配置文件 /etc/nginx/sites-enabled/default 文件中配置，内容如下：
server {
    # the port your site will be served on
    listen 80;
    # the domain name it will serve for
    server_name 127.0.0.1; # substitute your machine's IP address or DOMAIN
    charset utf-8;

    # send upstream error
    client_max_body_size 10M; # adjust to taste

    # Django media
    location /media {
        alias /home/ubuntu/.../media; # your Django project's media files - append as required
    }

    location /static {
        alias /home/ubuntu/.../static; # your Django project's static files - append as required
    }

    # Finally, send all non-media requests to the Django server.
    location / {
        include uwsgi_params; # the uwsgi params file you installed
        uwsgi_pass 127.0.0.1:8001;
    }
}

```

配置Nginx

7、收集

小礼物走一走，来简书关注我

赞赏支持

Django (/nb/25805946) 举报文章 © 著作权归作者所有



一许青衫一 (/u/8c6979d8ab4f)

写了 64885 字，被 6 人关注，获得了 27 个喜欢

(/u/8c6979d8ab4f)


+ 关注

喜欢 | 14



更多分享





下载简书 App ▶

随时随地发现和创作内容



(/apps/redirect?utm_source=note-bottom-click)

(/apps/redirect?utm_source=note-bottom-click)





登录后发表评论 (/sign_in?utm_source=desktop&utm_medium=not-signed-in-comments)

评论

智慧如你，不想发表一点想法 (/sign_in?utm_source=desktop&utm_medium=not-signed-in-nocomments-text) 咩~

被以下专题收入，发现更多相似内容


- 

我爱编程 (/c/7847442e0728?utm_source=desktop&utm_medium=notes-included-collection)
- 

Django (/c/7b1fae538656?utm_source=desktop&utm_medium=notes-included-collection)


新手的Flask+uwsgi+Nginx+Ubuntu部署过程 (/p/5b73444eb47d?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)

学习 Flask，写完一个 Flask 应用需要部署的时候，就想着折腾自己的服务器。根据搜索的教程照做，对于原理一知半解，磕磕碰碰，只要运行起来了，谢天谢地然后不再折腾了，到下一次还需要部署时，这样的...

 ershoucoder (/u/37c3c9744ed4?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)

Nginx + Django + uWSGI部署服务器详细解析 (/p/a13307242ca3?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)

本文基于：阿里云CentOS 7.4.1708, Python 3.6.1, Django 2.0.2, uWSGI 2.0.15, Nginx 1.12.2 知道怎么样部署之前我们要先弄清楚为什么？为什么要用Django + uWSGI + Nginx？所以更首先我们要知道...

 KenZhangCn (/u/8e971ded89ee?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)




(/p/e7979bdf225e?



(/apps/red
utm_sourc
banner-cl

utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendation
Python Web部署方式全汇总 (/p/e7979bdf225e?utm_campaign=maleski...

学过PHP的都了解，php的正式环境部署非常简单，改几个文件就OK，用FastCgi方式也是分分钟的事情。相比之下，Python在web应用上的部署就繁杂的多，主要是工具繁多，主流服务器支持不足。转载声明：本...

 有文化_0a9b (/u/7dd19402d450?


utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation

(/p/57a05dc698f6?



utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendation
《django by example》实践 上线运行 (/p/57a05dc698f6?utm_campaign=...


点我查看本文集的说明及目录。本文是《 django by example 》第十三章 上线运行 的内容。CH13 上线运行 上一章，我们为项目创建了 RESTful API 。本章，我们将学习如何实现以下功能： 配置生产环境 创建...

 学以致用123 (/u/4b9c0cc34415?

utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation

Python Web部署方式总结 (/p/0aece015976f?utm_campaign=maleskine...

学过PHP的都了解，php的正式环境部署非常简单，改几个文件就OK，用FastCgi方式也是分分钟的事情。相比之下，Python在web应用上的部署就繁杂的多，主要是工具繁多，主流服务器支持不足，在了解Python...

 chen_000 (/u/22621720d9dd?


utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation

(/p/d89c6a4f4566?



utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendation
最新版欧舒丹护手霜六支套装 (/p/d89c6a4f4566?utm_campaign=maleski...

¥360 里面包含樱花*1~牡丹花香*1~玫瑰*1~马鞭草*1~乳木果*2。各30ml。一盒就拥有这么多种香味的护手霜 还不快来一套！

 丹sir购 (/u/4dfa05b58792?

utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation

(/p/77f17f24ae91?



utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendation
居想网：华彬庄园中央空调安装实例鉴赏 (/p/77f17f24ae91?utm_campaig...

项目名称：华彬庄园金雄会所 项目介绍：华彬庄园位于北京市昌平区，占地6000亩，踞临长城脚下，是集体育运动、休闲度假、会议、婚庆、养生为一体的五星级豪华酒店。为社会精英人士打造非凡的尊贵体验，...

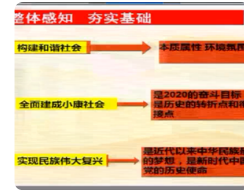


居想网 (/u/434a4fa931b3?

utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recom(https://maleskine.com/)

utm_source=

(/p/b2f938452189?



utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendation;

100节课第7节 (/p/b2f938452189?utm_campaign=maleskine&utm_conte...

展望祖国的未来 一起跟着老师学习吧。



扬帆起航18 (/u/74c94b7b452e?)

utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation

Shell Melody (/p/243726e49701?utm_campaign=maleskine&utm_cont...

我把时间的磁带倒回那个傍晚 又把海浪的声音录进那个贝壳 小小的橙色的它是有大海旋律 轻轻地触及着我
甜甜的思绪 送给你 我想让她在你心里呆得久一点 再久一点 你可不可以不把她丢弃



浓浓冬浅浅夏 (/u/1104cc05e126?)

utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation

(/p/377e8c06f7a6?



utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendation

恒泰金服权威解读未来供应链金融 (/p/377e8c06f7a6?utm_campaign=mal...

供应链金融是基于真实的交易背景开展的金融活动，对供应链的某个环节或全链条提供定制化的金融服务，通过整合信息、资金、物流等资源，达到提高资金使用效率、为各方创造价值和降低风险的作用。这种服...



句有料 (/u/57f2e35ce7d1?

utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation

