**宿舍管理系统之服务器部署**

**（张敏华）**

## 一、目的：

作为一个Web项目，在本地运行是远远不够，需要能被其他人通过外网访问，因此，需要将项目部署到服务器上，让项目在服务器上运行。

## 二、环境：

**2.1 云服务器信息：**

IP: 39.108.185.66

OS：ubuntu 14.04.5 LTS

**2.2 项目框架：**

Django：1.11.4

2.3 支持软件以版本号：

Python：3.4.

uWSGI: 2.0.15

Nginx: 1.4.6

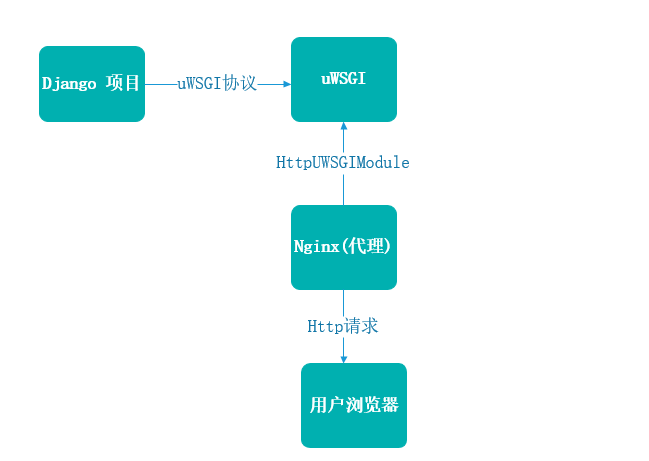
**2.3 Nginx和uWSGI介绍：**

Nginx是一款轻量级的Web服务器/反向代理服务器及电子邮件IMAP/POP3代理服务器，其特点是占有内存少，并发能力强。利用Nginx来部署服务器，主要目的是实现负载均衡。

uWSGI是一个Web服务器网关接口，它实现了WSGI协议、uwsgi、http等协议。Nginx中HttpUwsgiModule的作用是与uWSGI服务器进行交换。

**2.4 部署服务器后，用户获取网站资源的流程:**

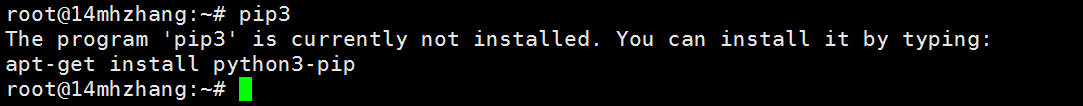
用户浏览器发送HTTP请求-->nginx(如果要获取的资源是静态文件，则直接返回静态文件)-->uWSGI(如果是动态请求，nginx利用socket和uwsgi进行交流和资源传送)-->uwsgi 从Django应用程序获取资源。流程图如下：



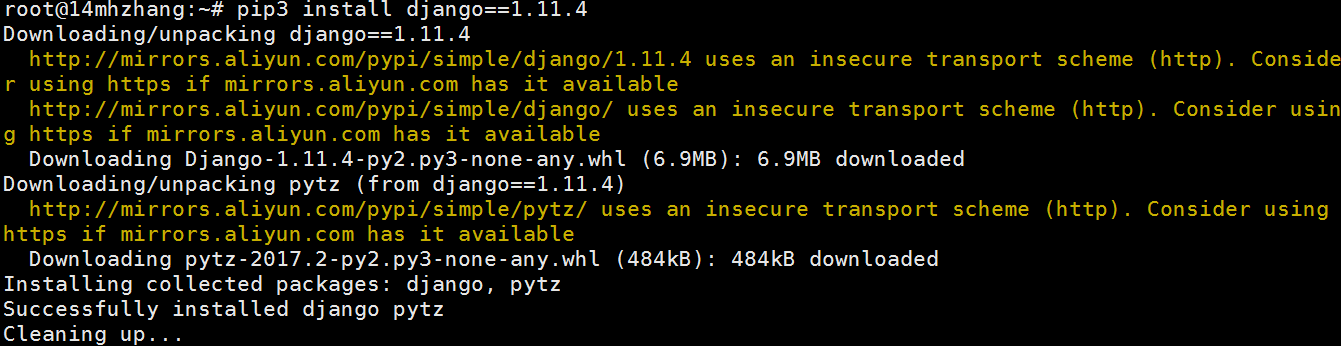
**三、服务器部署过程：**

**3.1** 安装python3，由于Ubuntu已经默认安装了python3，版本是python 3.4.3，所以这一步可省略。

**3.2**安装pip3



**3.3**安装django 1.11.4



**3.4 安装nginx:**



**3.5 安装uwsgi:**

$~ pip install uwsgi

测试是否安装成功:

新建test\_uwsgi.py文档，写入下面内容：

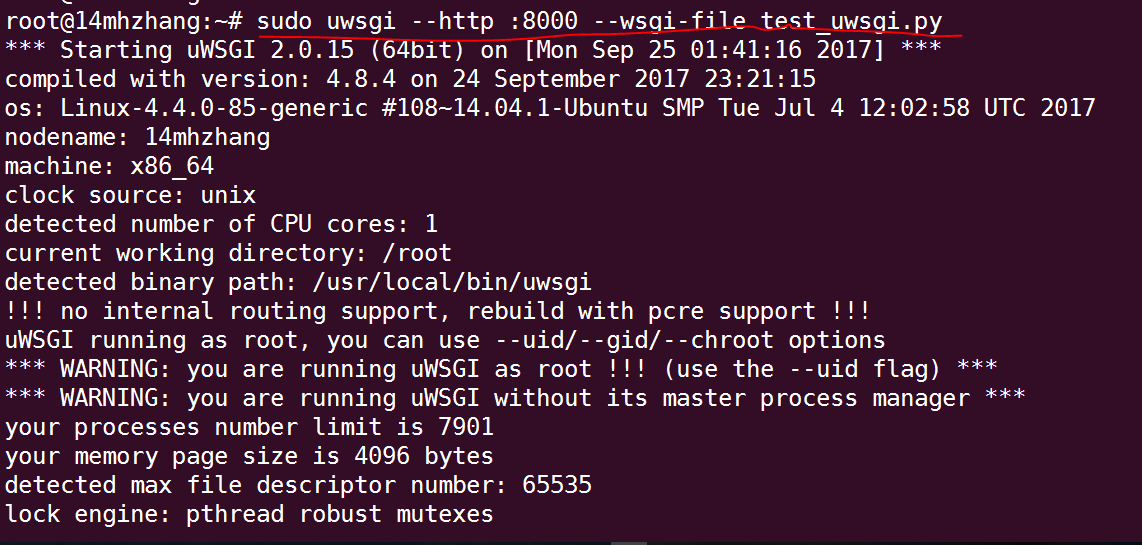
**def** **application**(env, start\_response):

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

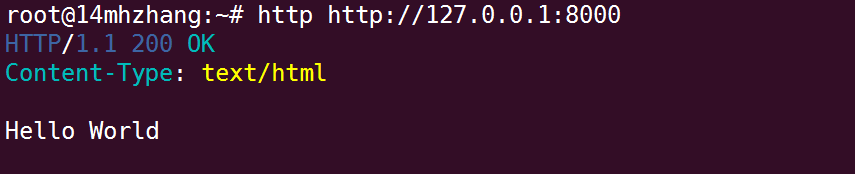
**return** [b"Hello World"]# python3

**# return** ["Hello World"]# python2

保存。然后用uwsgi运行该文件



测试8000端口

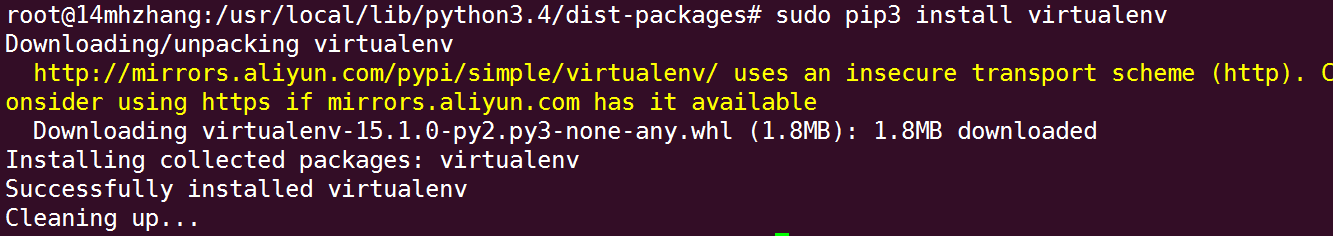


至此，成功安装了uWsgi。

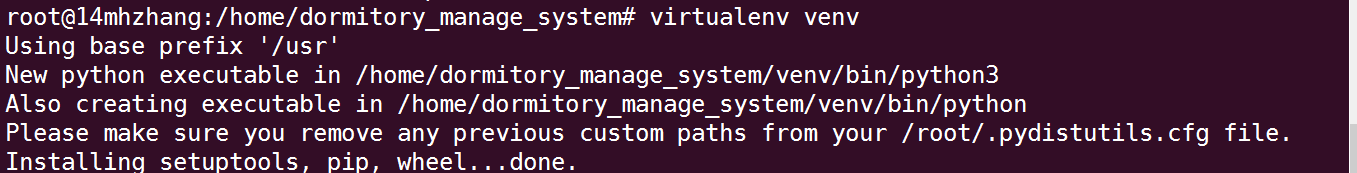
**3.6 搭建项目的虚拟环境**

因为ubuntu下有两个版本的python，为了项目在ubuntu上运行时拥有独立的运行环境，避免因为python版本不兼容问题而运行不成功，所以在服务器上搭建了Django项目的虚拟环境，主要是使用python3的virtualenv模块，虚拟环境搭建过程如下：

安装virtualenv：



创建虚拟环境:



激活虚拟环境:

$ ~ source /venv/bin/activate

在虚拟环境下安装Django1.11.4和PyMySQL模块，并利用Git克隆项目到服务器上。

**3.7 添加项目的uwsgi配置文件：**

在项目目录下，新建projectName\_uwsgi.ini配置文件，其内容如下：

#dor\_system\_uwsgi.ini file :

[uwsgi]

# Django-related settings

socket = :8000

# the base directory (full path)

chdir = /home/dormitory\_manage\_system/dormitory-manage-system

# Django s wsgi file

module = dor\_system.wsgi

# process-related settings

# master

master = true

# maximum number of worker processes

processes = 4

# ... with appropriate permissions - may be needed

# chmod-socket = 664

# clear environment on exit

vacuum = true

**3.8 修改nginx配置文件：**

Nginx的配置文件是/etc/nginx/nginx.cong，用vim打开nginx.conf，并添加下面内容

……

server {

listen 8000;

server\_name 127.0.0.1

charset UTF-8;

access\_log /var/log/nginx/myweb\_access.log;

error\_log /var/log/nginx/myweb\_error.log;

client\_max\_body\_size 75M;

location / {

include uwsgi\_params;

uwsgi\_pass 127.0.0.1:8000;

uwsgi\_read\_timeout 2;

}

location /static {

expires 30d;

autoindex on;

add\_header Cache-Control private;

alias /home/dormitory\_manage\_system/dor\_system/static/;

}

}

自此，服务器的部署结束。