# **Python3 Flask+nginx+Gunicorn部署（上）**

<https://blog.csdn.net/xudailong_blog/article/details/80490137>

前言：一般在本地运行flask项目通常是直接****python3 文件名.py****，然后打开：http://127.0.0.1:5000 查看代码结果

这次主要是记录flask在python3 环境结合nginx +gunicorn在服务器上进行项目的部署

####（一）运行环境：

* 虚拟机centos7
* python3 环境
* nginx
* gunicorn
* virtualenv

难点：nginx gunicorn的安装配置

#### **（二）nginx、gunicorn简介**

Nginx是一款[轻量级](https://baike.baidu.com/item/%E8%BD%BB%E9%87%8F%E7%BA%A7/10002835)的[Web](https://baike.baidu.com/item/Web/150564) 服务器/[反向代理](https://baike.baidu.com/item/%E5%8F%8D%E5%90%91%E4%BB%A3%E7%90%86)服务器及[电子邮件](https://baike.baidu.com/item/%E7%94%B5%E5%AD%90%E9%82%AE%E4%BB%B6)（IMAP/POP3）代理服务器，并在一个BSD-like 协议下发行。其特点是占有内存少，[并发](https://baike.baidu.com/item/%E5%B9%B6%E5%8F%91)能力强，事实上nginx的并发能力确实在同类型的网页服务器中表现较好，中国大陆使用nginx网站用户有：百度、[京东](https://baike.baidu.com/item/%E4%BA%AC%E4%B8%9C)、[新浪](https://baike.baidu.com/item/%E6%96%B0%E6%B5%AA)、[网易](https://baike.baidu.com/item/%E7%BD%91%E6%98%93)、[腾讯](https://baike.baidu.com/item/%E8%85%BE%E8%AE%AF)、[淘宝](https://baike.baidu.com/item/%E6%B7%98%E5%AE%9D)等

gunicorn是一个python Wsgi http server，只支持在Unix系统上运行，来源于Ruby的unicorn项目。Gunicorn使用prefork master-worker模型（在gunicorn中，master被称为arbiter），能够与各种wsgi web框架协作。

####（三）软件安装

（1）首先安装gunicorn

直接使用命令：pip install gunicorn

（2） 将gunicorn 加入到app.run（）中，

这里我在路径为：/home/flaskproject/flaskweb 下新建一个myweb.py 作为入口函数  
代码为：

from flask import Flask

app = Flask(\_\_name\_\_)

@app.route('/')

def index():

return 'hello world'

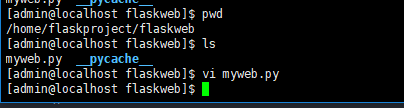
if \_\_name\_\_ == '\_\_main\_\_':

from werkzeug.contrib.fixers import ProxyFix

app.wsgi\_app = ProxyFix(app.wsgi\_app)

app.run(

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10

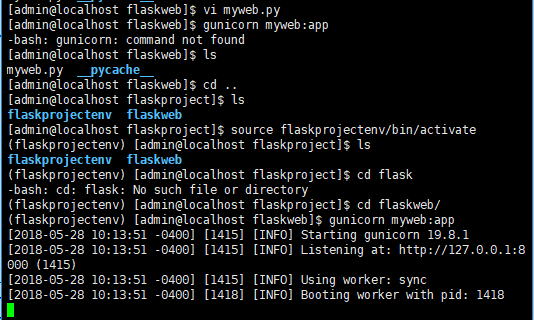


（3）用命令启动gunicorn

在myweb.py路径下，一定要记住是当前路径下！！  
方式一：

gunicorn myweb:app

* 1



python 虚拟环境的安装：

pip install virtualenv

* 1

然后一顿骚操作：

mkdir flaskproject

cd flaskproject

virtualenv flaskprojectenv

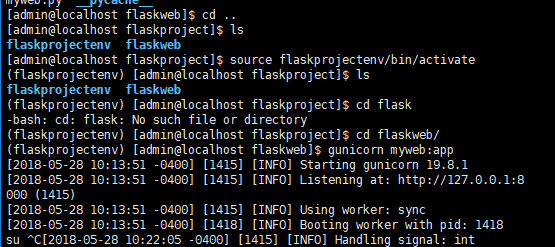
* 1
* 2
* 3
* 4
* 5

然后进行激活：

source flaskprojectenv/bin/activate

* 1

正如下图一样的骚操作，之前我也是很少用虚拟环境的，现在感觉虚拟环境并没有那么神秘，因为以前是很拒绝，不会用，不过这次是会用了，

退出虚拟环境的命令是：****deactivate****(这里我只是说一下)  


这时候我们本地服务器看一下是否运行起来（已经有hello world）：

[root@localhost flaskproject]# curl http://127.0.0.1:8000

hello world[root@localhost flaskproject]#

* 1
* 2

ctrl + c 停掉当前环境，我们使用第二种方式试一下

专门为处理高并发则要开多个进程和修改监听端口方式：

gunicorn -w 4 -b 127.0.0.1:8000 入口文件名:app

* 1

如下代码：

(flaskprojectenv) [admin@localhost flaskweb]$ gunicorn -w 4 -b 127.0.0.1:8000 myweb:app

[2018-05-28 10:57:11 -0400] [1813] [INFO] Starting gunicorn 19.8.1

[2018-05-28 10:57:11 -0400] [1813] [INFO] Listening at: http://127.0.0.1:8000 (1813)

[2018-05-28 10:57:11 -0400] [1813] [INFO] Using worker: sync

[2018-05-28 10:57:11 -0400] [1816] [INFO] Booting worker with pid: 1816

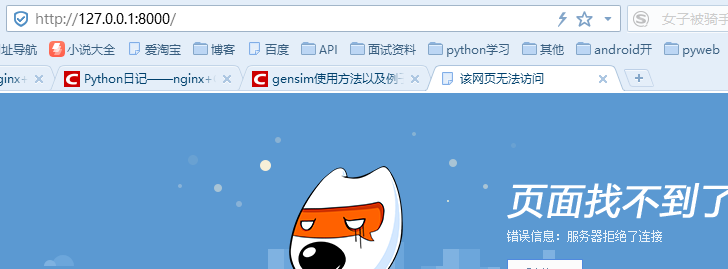
[2018-05-28 10:57:11 -0400] [1817] [INFO] Booting worker with pid: 1817

[2018-05-28 10:57:11 -0400] [1819] [INFO] Booting worker with pid: 1819

[2018-05-28 10:57:11 -0400] [1821] [INFO] Booting worker with pid: 1821

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8

本地服务器已经运行了，但是我们远程并不能进行访问（win10访问虚拟机）



说明端口号没有打开，这时候我们需要把防火墙，端口号什么的都进行设置一下，可以看一下这篇博客：[linux下nginx首次安装远程无法访问](https://blog.csdn.net/xudailong_blog/article/details/80468123)

主要是两个命令：

[root@localhost nginx-1.12.1] systemctl stop firewalld

[root@localhost nginx-1.12.1] systemctl stop iptalbes

* 1
* 2

接下来：我们进行nginx的配置安装与gunicorn的相结合并进行部署。