

[登录](#)

2019年05月04日 阅读 8226

[关注](#)

用uWSGI和Nginx部署Flask项目

概况

在开发过程中,我们一般直接用Python命令直接运行Flask程序。这样的运行只适合我们开发,方便我们调试。一旦程序部署到线上,这样运行的Flask程序性能会比较低。可以采用uWSGI+Nginx进行部署。

uWSGI

在部署之前,我们得先了解几个概念

- wsgi web应用程序之间的接口。它的作用就像是桥梁,连接在web服务器和web应用框架之间。
- uwsgi 是一种传输协议,用于定义传输信息的类型。
- uWSGI 是实现了uwsgi协议WSGI的web服务器。

部署

首先准备一个flask程序,名字run.py

```
from flask import Flask

app = Flask(__name__)

@app.route("/")
def index():
    return '部署'

if __name__ == '__main__':
    app.run()
```

[复制代码](#)

安装uWSGI

```
pip install uwsgi
```

[复制代码](#)

在项目的目录创建一个uwsgi的配置文件

```
[uwsgi]
#配合nginx使用
```

[复制代码](#)

```
socket = 127.0.0.1:8000
#项目路径 /Users/xiaoyuan/Desktop/flask_test
chdir      = 自己项目路径
#wsgi文件 run就是flask启动文件去掉后缀名 app是run.py里面的Flask对象
module     = run:app
#指定工作进程
processes  = 4
#主进程
master     = true
#每个工作进程有2个线程
threads = 2
#指的后台启动 日志输出的地方
daemonize  = uwsgi.log
#保存主进程的进程号
pidfile = uwsgi.pid
#虚拟环境环境路径
virtualenv = /Users/xiaoyuan/.virtualenvs/flask_test
```

然后配置Nginx, 找到Nginx的配置文件nginx.conf, 在http块下加上下面代码

```
server {
# 监听端口
listen 80;
# 监听ip 换成服务器公网IP
server_name 127.0.0.1;

#动态请求
location / {
include uwsgi_params;
uwsgi_pass 127.0.0.1:8000;
}
#静态请求
location /static {
alias /Users/xiaoyuan/Desktop/flask_test/static;

}
}
```

复制代码

启动

启动uWSGI

```
uwsgi --ini uwsgi.ini
```

复制代码

启动Nginx

```
/etc/init.d/nginx start
```

复制代码

然后访问Nginx监听的IP, 由于我在本地部署, 就直接访问http://127.0.0.1:80/



其他命令

- uwsgi重启

```
uwsgi --reload uwsgi.pid
```

复制代码

- uwsgi停止

```
uwsgi --stop uwsgi.pid
```

复制代码

欢迎关注我的公众号：



文章分类

后端

文章标签

Flask

相关推荐



