### 一、nginx+<mark>uwsgi</mark>+django

### 1、安装python3.6

```
1 安装依赖
2 yum -y install sqlite-devel
3 yum -y install make zlib zlib-devel gcc-c++ libtool openssl openssl-
   devel pcre pcre-devel git
4
5 #安装Python
6 wget https://www.python.org/ftp/python/3.6.1/Python-3.6.1.tgz
7 tar zxvf Python-3.6.1.tgz
8 cd Python-3.6.1
9 ./configure --prefix=/usr/local/python3
10 make && make install
11 #软连接
12 ln -s /usr/local/python3/bin/python3 /usr/bin/python3
13 ln -s /usr/local/python3/bin/pip3 /usr/bin/pip3
14
15 python3 -V
```

#### 2.dajngo 安装

```
1 下载地址 https://www.djangoproject.com/download/
2 tar zxvf Django-2.0.tar.gz
3 cd Django-2.0
4 python3 setup.py install
5
6 #测试
7 上传django项目
8 cd /root/day1_BookDeleteAndAdd
9 python3 manage.py runserver 0.0.0.0:80
10 访问http://127.0.0.1 可以访问
```

# 二、安装<mark>uwsgi</mark> web网关

```
1 wget http://projects.unbit.it/downloads/uwsgi-2.0.1.tar.gz
2 tar zxvf uwsgi-2.0.1.tar.gz
3
4 cd uwsgi-2.0.1
5 python3 uwsgiconfig.py --build
6 python3 uwsgiconfig.py --clean
8 cp -R /root/uwsgi-2.0.1 /usr/local/uwsgi
9 ln -s /usr/local/uwsgi/uwsgi /usr/bin/uwsgi
10
11 创建测试文件 vim test.py
12 def application(env, start_response):
       start_response('200 OK', [('Content-Type', 'text/html')])
13
14
      return [b"Hello World"]
15
16 #启动
17 /usr/local/uwsgi/uwsgi --http :9090 --wsgi-file /root/test.py
18 访问http://ip:9090 可以访问成功
```

# 三、<mark>uwsgi</mark>和django结合

```
1 #--chdir /root/day1_BookDeleteAndAdd django项目目录
2 #day1_BookDeleteAndAdd/wsgi.py day1....是程序目录
3 uwsgi --http:9090 --chdir /root/day1_BookDeleteAndAdd/ --wsgi-file day1_BookDeleteAndAdd/wsgi.py --master --processes 4 --threads 2 --stats 127.0.0.1:9192
4 #--stats 127.0.0.1:9192 不知道用处
5
6 访问http:ip:9090 可以访问成功
7 #如果在访问django项目时静态文件加载失败可以先忽略,在nginx配置中会指定静态文
```

#### 参数讲解

```
1 http: 协议类型和端口号
2
3 processes: 开启的进程数量
4
5 workers: 开启的进程数量,等同于processes(官网的说法是spawn the
  specified number ofworkers / processes)
6
7 chdir: 指定运行目录(chdir to specified directory before apps
  loading)
8
9 wsgi-file: 载入wsgi-file(load .wsgi file)
10 stats: 在指定的地址上,开启状态服务(enable the stats server on the
  specified address)
11
12 threads: 运行线程。由于GIL的存在,我觉得这个真心没啥用。
13
14 master : 允许主进程存在 (enable master process)
15
16 daemonize: 使进程在后台运行,并将日志打到指定的日志文件或者udp服务器
   (daemonize uWSGI)。实际上最常用的,还是把运行记录输出到一个本地文件上。
17
18 pidfile : 指定pid文件的位置,记录主进程的pid号
20 vacuum : 当服务器退出的时候自动清理环境,删除unix socket文件和pid文件(try
  to remove all of the generated file/sockets)
21
22 注意: -wsgi-file后面跟的是相对目录
```

4、<mark>uwsgi</mark>有两种启动方式,一种通过端口即上边这种方式,另一种通过配置.ini文件启动即下边用 到的方式

```
1 在django项目中/根目录下操作
```

```
3 cd /root/day1_BookDeleteAndAdd && mkdir -p
   /root/day1_BookDeleteAndAdd/logs
4 vim uwsgi.ini
5
  [uwsgi]
6
7 #uwsgi启动端口
8 socket = 127.0.0.1:9090
10 chdir=/root/day1_BookDeleteAndAdd
11 module=day1_BookDeleteAndAdd.wsgi
12 master = true
13 processes=2
14 threads=2
15 max-requests=2000
16 chmod-socket=664
17 vacuum=true
18 # 日志路径
19 daemonize = /root/day1_BookDeleteAndAdd/logs/uwsgi.log
```

## 四、安装nginx

### 1、下载安装nginx

```
1 yum -y install openssl openssl-devel pcre pcre-devel #省略
2 wget https://nginx.org/download/nginx-1.12.2.tar.gz --no-check-certificate
3
4 tar zxvf nginx-1.9.9.tar.gz
5 cd nginx-1.9.9
6 ./configure --prefix=/usr/local/nginx --with-pcre --with-http_stub_status_module --with-http_ssl_module --with-http_gzip_static_module --with-http_sub_module --with-cc=/usr/bin/gcc
7 make && make install
8
9 useradd -s /sbin/nologin -M www
10 cp -a /usr/local/nginx/sbin/nginx /etc/init.d/
11 cd /usr/local/nginx/conf
12 cp -f nginx.conf nginx.conf_bak
```

```
13
14 sed -i 's/#user nobody/user www/g' nginx.conf #改启动用户
15 sed -i 's/worker_processes 1/worker_processes 8/g' nginx.conf #自定义 线程数
16 /usr/local/nginx/sbin/nginx #启动nginx #/etc/init.d/nginx
17 #/etc/init.d/nginx -s stop 关闭
18
19 访问测试http://ip 默认80端口,可以访问
```

# 五、nginx+uwsgi+django三者结合

#### 1、配置nginx

```
1 cp -a /usr/local/nginx/conf/nginx.conf /tmp
2 mkdir -p /var/log/nginx/guolm
3
4
  vim /usr/local/nginx/conf/nginx.conf #如下全部覆盖
5
6
7 user www;
8 worker_processes 8;
9 error_log logs/error.log;
         logs/nginx.pid;
10 pid
11
12 worker_rlimit_nofile 65535;
13 events {
14
          use epoll;
          worker_connections 65535;
15
16 }
17 http {
18
       include
                mime.types;
19
       default_type application/octet-stream;
20
21
       log_format main '$remote_addr - $remote_user [$time_local]
   "$request" '
                         '"$status" $body_bytes_sent "$http_referer" '
22
                         '"$http_user_agent" "$http_x_forwarded_for"';
23
       log_format wireless '$remote_addr - $remote_user [$time_local]
24
```

```
"$request" '
25
                            '"$status" $body_bytes_sent "$http_referer" '
26
                            '"$http_user_agent" $http_x_forwarded_for
   "$request_time"';
27
       sendfile
                       on;
       keepalive_timeout 65;
28
       include vhosts.conf;
29
       include upstream.conf;
30
31 }
32
33 vim /usr/local/nginx/conf/vhosts.conf
34 #访问自己
35 server {
36
           listen
                        80;
           server_name localhost;
37
           error log /var/log/nginx/guolm/error.log;
38
           access_log /var/log/nginx/guolm/access.log wireless;
39
40
           location / {
41
               include uwsgi_params;
42
43
               #uwsgi pass 127.0.0.1:9090;
44
               uwsgi_pass guolm;
45
               root
                      html;
46
               index index.html index.htm;
47
           }
48
           error_page 500 502 503 504 /50x.html;
           location = /50x.html {
49
50
               root
                      html;
51
           }
         }
52
53
54 vim /usr/local/nginx/conf/upstream.conf
55 #9090 是uwsgi地址
56 upstream guolm {
57
     server 127.0.0.1:9090;
58 }
59
```

#### #下边必须操作,否则后台管理admin的静态资源会无法访问

```
1 1)第一种方法:找到django-admin静态文件的目录
2 >>> import django
```

```
3 >>> django.__file__
4 '/usr/local/python3/lib/python3.6/site-packages/Django-2.0.4-
    py3.6.egg/django/__init__.py'
5 找到根目录之后我需要的绝对路径为(不同版本可能不一样,根据自己实际情况来)
6 /usr/local/python3/lib/python3.6/site-packages/Django-2.0.4-
    py3.6.egg/django/contrib/admin/static
7 把static 拷贝到自己定义的static目录,一般是在项目的根目录下,或者应用的根目录下,
8
9 2)第二种方法,建立自己的静态文件夹
10 修改settings,建立自己的文件夹
11 STATIC_ROOT = "/root/day1_BookDeleteAndAdd/static"
12 运行下面命令把相关文件copy到这个目录
13 python manage.py collectstatic
```

### 3、启动<mark>uwsgi</mark>和nginx

```
1 #uwsgi 启动
2 uwsgi --ini /root/day1_BookDeleteAndAdd/uwsgi.ini
3 #报错 wsgi: error while loading shared libraries: libpcre.so.1: cannot open shared object file: No such file or directory
4 #解決: cd /lib64 && ln -s libpcre.so.0.0.1 libpcre.so.1
5 #nginx启动
6 /usr/local/nginx/sbin/nginx #(存在就重启)
7
8 访问 http://192.168.87.128
9 #遇到的问题访问时报错, /usr/local/nginx/logs/error.log
10 #先尝试把djnago项目目录加权限 chown www:www /root/day1_BookDeleteAndAdd
11 #发现还是不行 后来把项目的上级目录也加www权限
```