**Mysql安装**

http://blog.csdn.net/zhangyong329/article/details/52785455

**Python安装**

1.yum install zlib-devel bzip2-devel openssl-devel ncurses-devel sqlite-devel readline-devel tk-devel mysql mysql-devel gcc

yum -y install make zlib zlib-devel gcc-c++ libtool openssl openssl-devel

cd /usr/local/src

wget https://www.python.org/ftp/python/3.6.0/Python-3.6.0.tgz

tar -zxf /usr/local/src/Python-3.6.0.tgz && cd Python-3.6.0

mkdir /usr/local/python3

./configure --prefix=/usr/local/python3

make && make altinstall

# 关键一步，建立链接

mv /usr/bin/python /usr/bin/python2.7.7

ln -s /usr/local/python3/bin/python3.6 /usr/bin/python # 可直接使用 "python"

ln -s /usr/local/python3/bin/pip3.6 /usr/bin/pip # 可直接使用 "pip"

# 最后测试下

python

pip

2.vim /usr/bin/yum

将开头的python改成python2.7

vim /usr/libexec/urlgrabber-ext-down

将/usr/bin/python改为/usr/bin/python2.7

**Pip插件安装**

pip install Django

pip install mysqlclient

pip install uwsgi

ln -s /usr/local/python3/bin/uwsgi /usr/bin/uwsgi #建立软连接

pip install pyopenssl ndg-httpsclient pyasn1

**uWsgi 部署**

新建test.py

uwsgi --http :8000 --wsgi-file test.py

内容：

def application(env, start\_response):

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

#return ["Hello World"] # python2

return [b"Hello World"] # python3

本地打开页面localhost:8000，可以看到Hello World

**Nginx安装部署**

cd /usr/local/src

wget http://ftp.exim.llorien.org/pcre/pcre-8.35.zip

unzip pcre-8.35.zip

cd pcre-8.35

./configure --prefix=/usr/local/pcre-8.35

make && make install

pcre-config --version

wget <http://nginx.org/download/nginx-1.6.2.tar.gz>

tar zxvf nginx-1.6.2.tar.gz

cd nginx-1.6.2

./configure --prefix=/usr/local/nginx --with-http\_stub\_status\_module --with-http\_ssl\_module --with-pcre=/usr/local/src/pcre-8.35

make

make install

/usr/local/nginx/sbin/nginx -v

ln -s /usr/local/nginx/sbin/nginx /usr/bin/nginx

nginx -v

到此，nginx安装完成。

**Nginx+uwsgi+Django部署**

接下来，我们要将三者结合起来。首先罗列一下项目的所需要的文件：

myweb/

├── manage.py

├── myweb/

│ ├── \_\_init\_\_.py

│ ├── settings.py

│ ├── urls.py

│ └── wsgi.py

└── myweb\_uwsgi.ini

在我们通过Django创建myweb项目时，在子目录myweb下已经帮我们生成的 wsgi.py文件。所以，我们只需要再创建myweb\_uwsgi.ini配置文件即可，当然，uwsgi支持多种类型的配置文件，如xml，ini等。此处，使用ini类型的配置。

# myweb\_uwsgi.ini file

[uwsgi]

# Django-related settings

socket = :8000

# the base directory (full path)

chdir = /home/fnngj/pydj/myweb

# Django s wsgi file

module = myweb.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

buffer-size = 36384

接下来，切换到myweb项目目录下，通过uwsgi命令读取myweb\_uwsgi.ini文件启动项目。

uwsgi --ini myweb\_uwsgi.ini

再接下来要做的就是修改nginx.conf配置文件

……

server {

listen 8099;

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/fnngj/pydj/myweb/static/;

}

}

……

**上传文件**

yum install lrzsz

rz