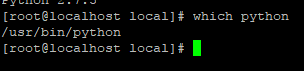
一. 安装python3.6

1.查看是否已经安装Python

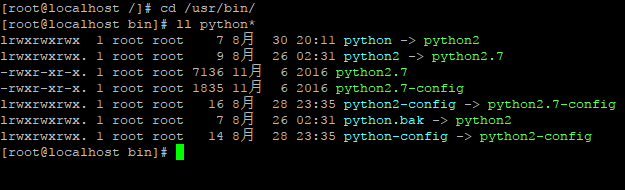
CentOS 7.2 默认安装了python2.7.5 因为一些命令要用它比如yum 它使用的是python2.7.5。

使用 python -V 命令查看一下是否安装Python

然后使用命令 which python 查看一下Python可执行文件的位置



可见执行文件在/usr/bin/ 目录下，切换到该目录下执行 ll python\* 命令查看



python 指向的是python2.7

因为我们要安装python3版本，所以python要指向python3才行。

目前还没有安装python3，先备份,备份之前先安装相关包，用于下载编译python3

提前安装好系统依赖包：

Centos:yum install openssl-devel bzip2-devel expat-devel gdbm-devel readline-devel sqlite-devel gcc gcc-c++ openssl-devel libffi-devel python-devel mariadb-devel

上一步执行完，就可以了，如果有异常就运行下面的命令

yum install zlib-devel bzip2-devel openssl-devel ncurses-devel sqlite-devel readline-devel tk-devel gcc make

不能忽略相关包，我之前就没有安装readline-devel导致执行python模式无法使用键盘的上下左右键；

然后备份

mv python python.bak

2.开始编译安装python3

去官网下载编译安装包或者直接执行以下命令下载

wget <https://www.python.org/ftp/python/3.6.2/Python-3.6.2.tar.xz>

解压

tar -xvJf Python-3.6.2.tar.xz

切换进入

cd Python-3.6.2

编译安装

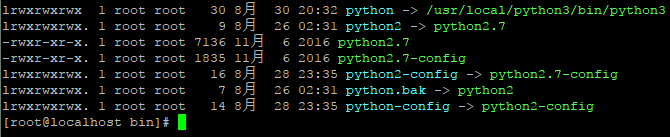
./configure prefix=/usr/local/python3

make && make install

安装完毕，/usr/local/目录下就会有python3了

因此我们可以添加软链到执行目录下/usr/bin

ln -s /usr/local/python3/bin/python3 /usr/bin/python



可以看到软链创建完成

测试安装成功了没，执行

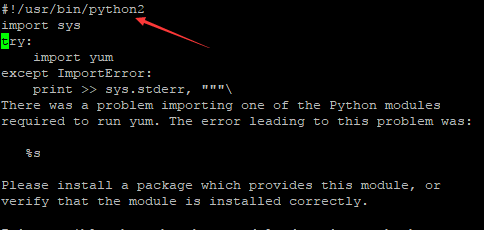
python -V 看看输出的是不是python3的版本

执行python2 -V 看到的就是python2的版本

因为执行yum需要python2版本，所以我们还要修改yum的配置，执行：

vi /usr/bin/yum

把#! /usr/bin/python修改为#! /usr/bin/python2



同理 vi /usr/libexec/urlgrabber-ext-down 文件里面的#! /usr/bin/python 也要修改为#! /usr/bin/python2

这样python3版本就安装完成；同时python2也存在

python -V 版本3

python2 -V 版本2

3.配置pip

Python3装完后，默认已经安装了pip，此时只要配置下软链接即可使用pip工具：

ln -s /usr/local/python3/bin/pip3 /usr/bin/pip

​

二. 安装maridb（与mysql功能一致）

1. 安装

sudo yum install mariadb-server

2. 启动， 重启

sudo systemctl start mariadb

sudo systemctl restart mariadb

3. 设置bind-ip

vim /etc/my.cnf

在 [mysqld]:

下面加一行

bind-address = 0.0.0.0

4. 设置外部ip可以访问

先进入mysql才能运行下面命令:

mysql 直接进入就行

GRANT ALL PRIVILEGES ON \*.\* TO 'root'@'%' IDENTIFIED BY '123456' WITH GRANT OPTION;

FLUSH PRIVILEGES

5. 设置腾讯云的对外端口（云服务器安全组中开放3306等端口）

6. 安装mysqlclient出问题

centos 7：

yum install python-devel mariadb-devel -y

pip install mysqlclient

3. 安装nginx

详细步骤见：<https://www.digitalocean.com/community/tutorials/how-to-install-nginx-on-centos-7>

sudo yum install epel-release

sudo yum install nginx

运行nginx的命令：

sudo /usr/sbin/nginx（权限最高建议使用）

这里需要注意 一定是直接用nginx命令启动， 不要用systemctl启动nginx不然会有权限问题

sudo systemctl start nginx 启动

sudo service nginx start 启动

sudo service nginx restart 重启

sudo pkill -f -9 nginx 彻底杀死nginx进程（建议使用）

sudo service nginx stop 停止

sudo /usr/sbin/nginx -s stop

输入服务器ip，即可看到nginx欢迎页面，代表nginx已安装成功且已启动

4. 安装virtualenvwrapper

yum install python-setuptools python-devel

pip install virtualenvwrapper

编辑.bashrc文件

export WORKON\_HOME=$HOME/.virtualenvs

source /usr/local/bin/virtualenvwrapper.sh

重新加载.bashrc文件

source ~/.bashrc

新建虚拟环境

mkvirtualenv mysite

进入虚拟环境

workon mysite

安装pip包

我们可以通过 pip freeze > requirements.txt 将本地的虚拟环境安装包相信信息导出来

然后将requirements.txt文件上传到服务器之后运行：

pip install -r requirements.txt

安装依赖包

5. 安装uwsgi

留意：uwsgi要安装两次，先在系统里安装一次，然后进入对应的虚拟环境安装一次。

系统里安装方法：yum intall uwsgi

Uwsgi --version

Workon 进入虚拟环境 pip install uwsgi

给uwsgi建立软链接，方便使用

ln -s /usr/local/python3/bin/uwsgi /usr/bin/uwsgi

7. 配置nginx

新建在项目路径下，新增nginx配置文件，/root/projects/mysite/guo.conf

# the upstream component nginx needs to connect to

upstream django {

# server unix:///path/to/your/mysite/mysite.sock; # for a file socket

server 127.0.0.1:8001; # for a web port socket (we'll use this first)

}

# configuration of the server

server {

# the port your site will be served on

listen 80;

# the domain name it will serve for

server\_name 148.70.221.132 ; # substitute your machine's IP address or FQDN

charset utf-8;

# max upload size

client\_max\_body\_size 75M; # adjust to taste

# Django media

location /media {

alias /root/projects/mysite/media; # 指向django的media目录

}

location /static {

alias /root/projects/mysite/static; # 指向django的static目录

}

# Finally, send all non-media requests to the Django server.

location / {

uwsgi\_pass django;

include /root/projects/mysite/uwsgi\_params; # the uwsgi\_params file you installed

}

}

注意：如果使用系统root账号时，nginx 可能静态文件访问不到 出现403错误，应在进入nginx.conf个文件下，将文件中的 user nginx; 改为user root；获得最高权限，然后执行

sudo pkill -f -9 nginx

sudo /usr/sbin/nginx

重启nginx即可

8. 将该配置文件加入到nginx的启动配置文件中

sudo ln -s /root/projects/mysite/guo.conf /etc/nginx/conf.d/

9. 拉取所有需要的static file 到同一个目录（很重要否则上线时，部署Django项目时，Admin后台管理CSS样式丢失

）

在django的setting文件中，添加下面内容：

STATIC\_URL = '/static/'

STATIC\_URL = “/static”

STATIC\_ROOT = os.path.join(BASE\_DIR, "static/")

如果有以下代码，请注释掉，负责上线时，django admin页面会找不到静态文件

STATICFILES\_DIRS = [  
 os.path.join(BASE\_DIR, "static"),  
 ]

打开项目/urls.py 添加如下内容：

url(r'^static/(?P<path>.\*)$', serve, {'document\_root': settings.STATIC\_ROOT}),

运行命令

python manage.py collectstatic

10. 通过配置文件启动uwsgi

新建uwsgi.ini,放在项目根目录，我的项目目录为 /root/projects/mysite/uwsgi.ini

此外：将uwsgi\_params文件也放到项目根目录/root/projects/mysite/uwsgi\_params

Uwsgi.ini内容如下：

# mysite\_uwsgi.ini file

[uwsgi]

# Django-related settings

# the base directory (full path)

#项目路径

chdir = /root/projects/mysite

# Django's wsgi file

#项目模块名

module = mysite.wsgi

# the virtualenv (full path)

# process-related settings

# master

master = true

# maximum number of worker processes最大进程数

processes = 10

# the socket (use the full path to be safe

#注意要与nginx中的socket地址一致

socket = 127.0.0.1:8001

# ... with appropriate permissions - may be needed

# chmod-socket = 66

# clear environment on exit

vacuum = true

#虚拟环境路径

virtualenv = /root/.virtualenvs/mysite

注：

chdir： 表示需要操作的目录，也就是项目的目录

module： wsgi文件的路径

processes： 进程数

virtualenv：虚拟环境的目录

注意要在虚拟环境中，启动uwsqi

workon mysite

uwsgi -i /root/projects/mysite/uwsgi.ini &

或者在虚拟环境中进入，项目根目录执行（即为有uwsgi文件的路径下）sudo uwsgi uwsgi.ini

杀死uwsgi进程

查看进程

ps -ef|grep uwsgi

sudo pkill -f -9 uwsgi(推荐使用)

sudo killall -9 uwsgi

12.

1、安装supervisor

supervisor管理uwsgi

在系统环境,不是虚拟环境,安装supervisor  
supervisor不支持3.x,系统的环境是2.7

注意，之前我们安装python3时，通过软连接 ln -s /usr/local/python3/bin/python3 /usr/bin/python 将python指向了python3

会出现以下错误

pkg\_resources.DistributionNotFound: The 'supervisor==3.2.0' distribution was

现在应为supervisor只支持python2，需要将python 指向Python2

rm -rf /usr/bin/python 删除原来的软链接

ln -s /usr/bin/python2 /usr/bin/python

通过

yum install supervisor（推荐使用）

安装 supervisor

也可以通过

pip install supervisor

卸载软件包

Yum remove supervisor

Pip uninstall supervisor

### 生成配置文件，且放在/etc目录下

echo\_supervisord\_conf > /etc/supervisord.conf

### 为了不将所有新增配置信息全写在一个配置文件里，这里新建一个文件夹，每个程序设置一个配置文件，相互隔离

mkdir /etc/supervisord.d/

### 修改配置文件

vim /etc/supervisord.conf

### 加入以下配置信息

[include]

files = /etc/supervisord.d/\*.conf

### 在supervisord.conf中设置通过web可以查看管理的进程，加入以下代码（默认即有，取消注释即可）

[inet\_http\_server]

port=9001

username=user

password=123

启动supervisord

supervisord -c /etc/supervisord.conf

查看supervisor是否启动成功

ps -ef|grep supervisord

将supervisor配置为开机自启动服务

见博客<https://blog.csdn.net/donggege214/article/details/80264811>

<https://my.oschina.net/u/2342969/blog/2986173>

现在通过 <http://ip:9001/> 就可以查看supervisor的web界面了(默认用户名及密码是user和123)，当然目前还没有加入任何监控的程序。

在/etc/supervisord.d/目录下新建一个mysite.conf的文件,内容如下

# 程序的名字，在supervisor中可以用这个名字来管理该程序,随便起

[program:uwsgi]

# 指定运行用户

user = root

# 启动程序的命令

command = /root/.virtualenvs/mysite/bin/uwsgi --ini /root/projects/mysite/uwsgi.ini

# 项目的目录

directory = /root/projects/mysite

# 开始的时候等待多少秒

startsecs = 0

# 停止的时候等待多少秒

stopwaitsecs = 0

# 设置改程序是否虽supervisor的启动而启动

autorstart = true

# 程序挂了是否需要重新将其启动

autorestart = true

# 是否将程序错误信息重定向到文件

redirect\_stderr=true

# 输出的log文件(log目录需要手动创建)

stdout\_logfile = /root/projects/mysite/supervisord.log

# 输出的错误文件(log目录需要手动创建)

stderr\_logfile = /root/projects/mysite/supervisord.err

[program:nginx]

# 指定运行用户

user = root

# 启动程序的命令

command=/usr/sbin/nginx -g 'daemon off;'

# 项目的目录

directory = /root ;

# 开始的时候等待多少秒

startsecs = 0

# 停止的时候等待多少秒

stopwaitsecs = 0

# 设置改程序是否虽supervisor的启动而启动

autorstart = true

[supervisord]

# log的级别

loglevel = info

重新启动supervisord，或者重新加载配置文件：

supervisorctl reload

### 或者

supervisorctl -c /etc/supervisord.conf

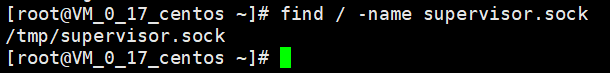
执行上述命令时可能会出现 Error: Another program is already listening on a port that one of our HTTP servers is configured touse. Shut this program down first before starting supervisord.  
For help, use /usr/bin/supervisord -h

解决办法：

find / -name supervisor.sock

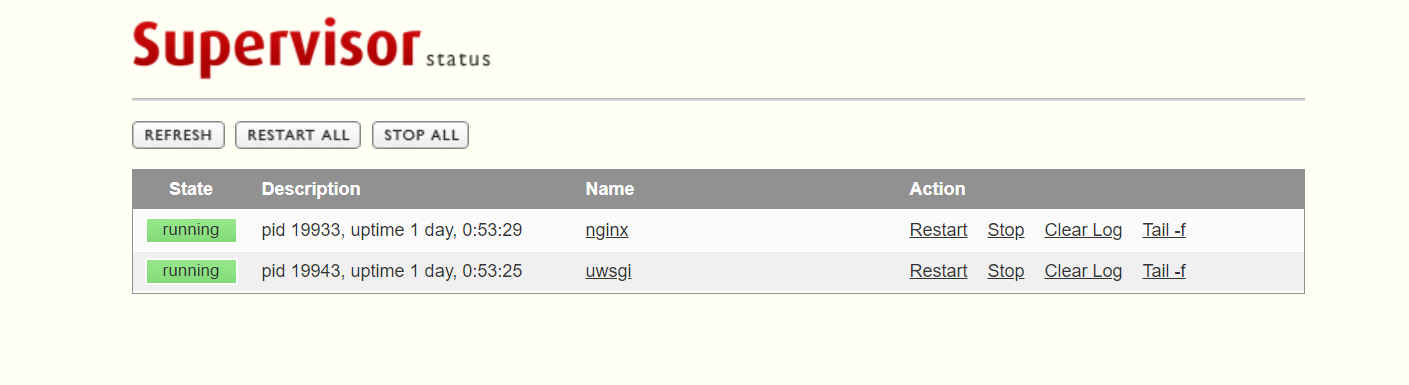
查找supervisor.sock的位置，然后删除软连接，

这个错误的原因就是supervisor.sock这个文件会被系统自动删除或者其它原因不存在了，删除软连接就可以了。  
supervisor.sock生成的位置可以去supervisor的配置文件中找到

unlink /tmp/supervisor.sock

打开浏览器,输入IP:9001  
输入账号和密码  
即可看到supervisor的web管理界面

如下图，监控了nginx和uwsgi两个程序



访问部署的网站

<http://你的ip地址/> 即可查看到你的网站