Flask应用部署

2019年8月4日　　21:15

服务器：阿里云云翼计划轻量应用服务器，1核2G，40G硬盘

系统：CentOS7.3

环境：Python3.7+MySQL+Gunicorn+Nginx+Supervisor

安装Python3.7

1. 安装依赖包

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

2. 再执行安装一个工具，不安装会导致pip安装失败

```
yum install libffi-devel -y
```

3. 下载python3.7的包，有两种方法，一种是从官网下载，然后上传到CentOS，另一种可以通过CentOS的wget命令直接下载。

- 官网下载：

  打开python的官方网站：https://www.python.org/，找到python3.7，下载Gzipped source tarball的包，然后使用scp命令或者其他工具将python包上传到服务器

- wget命令：

  ```
  wget https://www.python.org/ftp/python/3.7.0/Python-3.7.0.tgz
  ```

4. 安装python3.7，依次执行下列代码即可

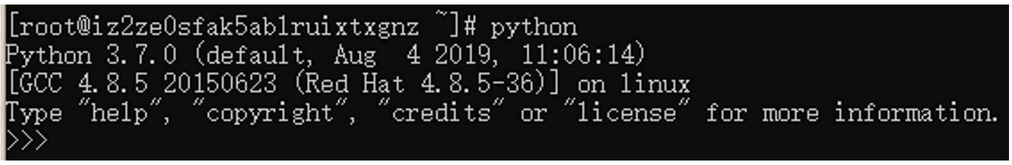
```
tar -zxvf Python-3.7.0.tgz
cd Python-3.7.0
./configure
make&&make install
```

5. 配置环境变量（python和pip）

```
mv /usr/bin/python /usr/bin/python.bak
ln -s /usr/local/bin/python3 /usr/bin/python
mv /usr/bin/pip /usr/bin/pip.bak
ln -s /usr/local/bin/pip3 /usr/bin/pip
```

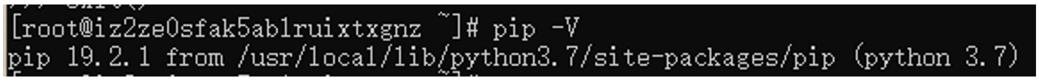
6. 验证

- 验证python，如果显示的版本号是3.7.0，就代表安装成功了（CentOS会自带一个python2.7）



```
[root@iz2ze0sfak5ab1ruixtxgnz ~]# python
Python 3.7.0 (default, Aug  4 2019, 11:06:14)
[GCC 4.8.5 20150623 (Red Hat 4.8.5-36)] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

- 验证pip，如果显示的版本号是10.0.1（下面的pip是更新之后的），以及最后python版本是3.7，就代表安装成功了



```
[root@iz2ze0sfak5ab1ruixtxgnz ~]# pip -V
pip 19.2.1 from /usr/local/lib/python3.7/site-packages/pip (python 3.7)
```

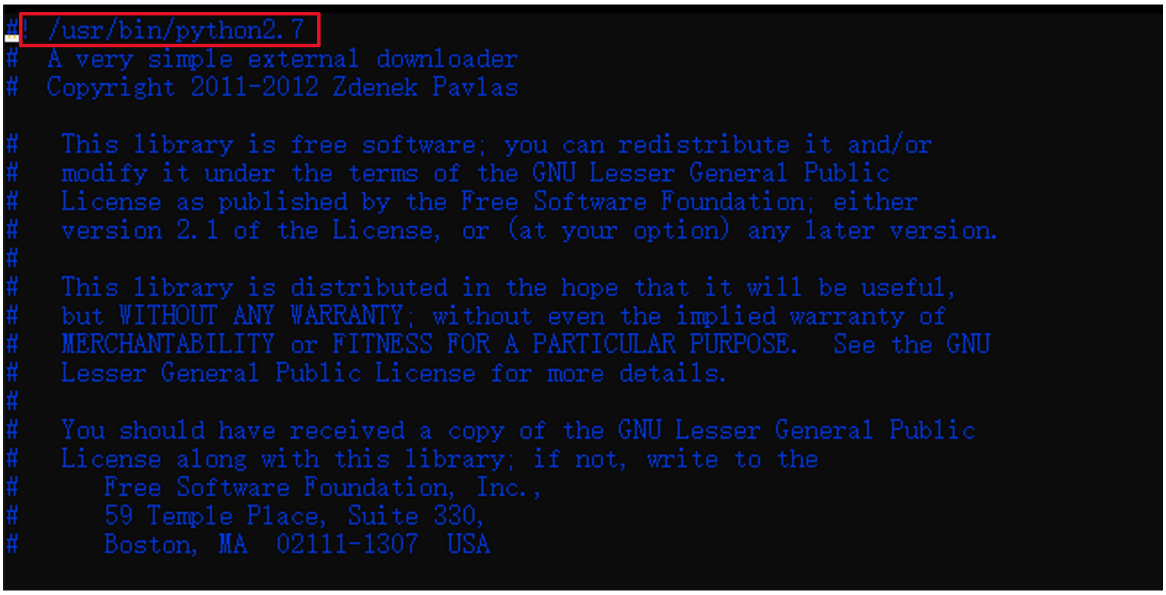
- 如果上述两个都没问题，就代表安装成功了！

7. 配置yum

这时候会发现yum不能用了，因为yum是依赖python2.7，我们把python改成3.7了，自然就不好使了，只要修改一下yum里面的依赖即可

```
vim /usr/libexec/urlgrabber-ext-down
```

打开以后，找到一个/usr/bin/python的，后面加上2.7就可以了，也就是/uer/bin/python2.7



```
#! /usr/bin/python2.7
#  A very simple external downloader
#  Copyright 2011-2012 Zdenek Pavlas

#   This library is free software; you can redistribute it and/or
#   modify it under the terms of the GNU Lesser General Public
#   License as published by the Free Software Foundation; either
#   version 2.1 of the License, or (at your option) any later version.
#
#   This library is distributed in the hope that it will be useful,
#   but WITHOUT ANY WARRANTY; without even the implied warranty of
#   MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
#   Lesser General Public License for more details.
#
#   You should have received a copy of the GNU Lesser General Public
#   License along with this library; if not, write to the
#      Free Software Foundation, Inc.,
#      59 Temple Place, Suite 330,
#      Boston, MA  02111-1307  USA
```

然后输入

```
vim /usr/bin/yum
```

做法和上面一样，改成python2.7就可以了

然后，我们的python开发环境就配置成功了！

安装MySQL

1. 下载mysql的repo源

```
wget http://repo.mysql.com/mysql-community-release-el7-5.noarch.rpm
```

2. 安装mysql-community-release-el7-5.noarch.rpm包

```
rpm -ivh mysql-community-release-el7-5.noarch.rpm
```

3. 安装mysql

```
yum install mysql-server
```

4. 重启mysql

```
service mysql restart
```

5. 登录mysql

```
mysql -u root
```

6. 修改密码

```
set password for 用户名@localhost = password('新密码');
```

用户名和密码填自己的

7. 修改字符集

- 查看字符集

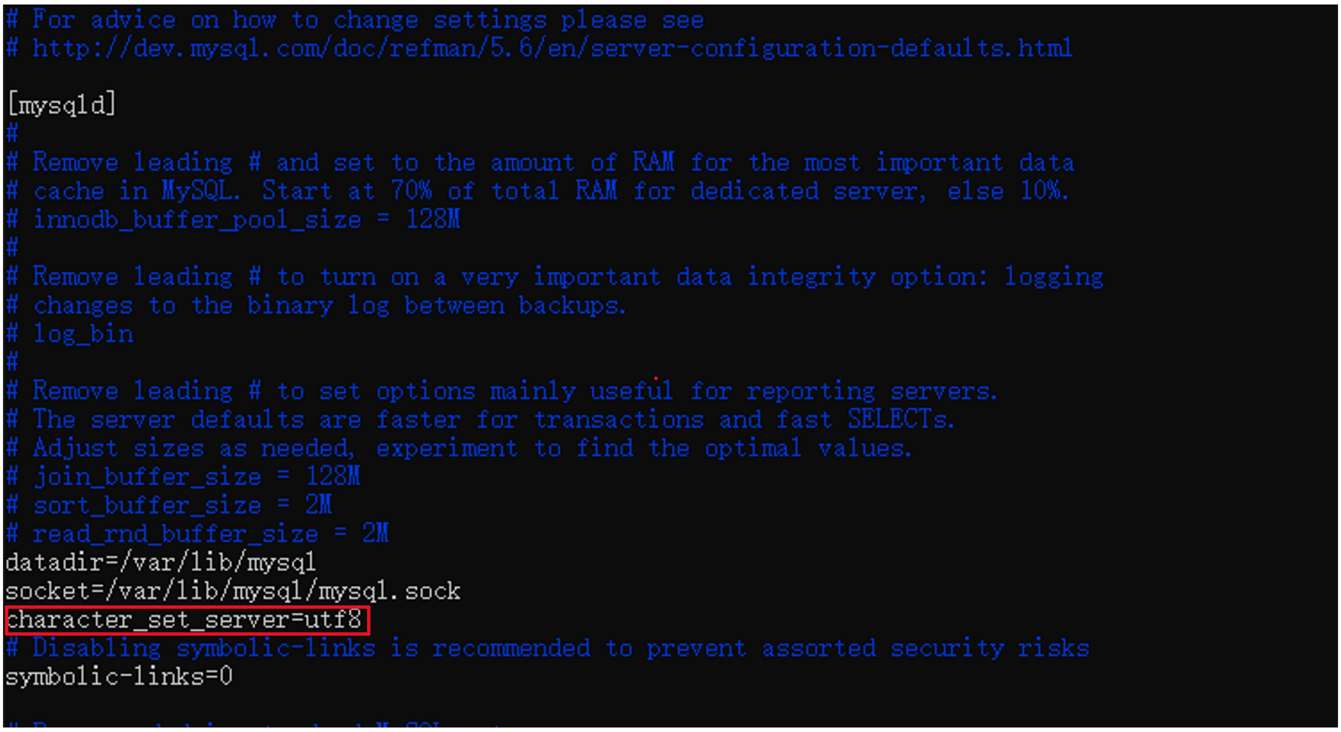
```
show variables like 'character%';
```

character_set_server的默认字符集为latin1

- 修改字符集

```
vim /etc/my.cnf
```

在[mysqld]字段里添加character_set_server=utf8



```
# For advice on how to change settings please see
# http://dev.mysql.com/doc/refman/5.6/en/server-configuration-defaults.html

[mysqld]
#
# Remove leading # and set to the amount of RAM for the most important data
# cache in MySQL. Start at 70% of total RAM for dedicated server, else 10%.
# innodb_buffer_pool_size = 128M
#
# Remove leading # to turn on a very important data integrity option: logging
# changes to the binary log between backups.
# log_bin
#
# Remove leading # to set options mainly useful for reporting servers.
# The server defaults are faster for transactions and fast SELECTs.
# Adjust sizes as needed, experiment to find the optimal values.
# join_buffer_size = 128M
# sort_buffer_size = 2M
# read_rnd_buffer_size = 2M
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
character_set_server=utf8
# Disabling symbolic-links is recommended to prevent assorted security risks
symbolic-links=0
```

- 重启数据库后再次查看字符集

  ```
  show variables like 'character%';
  ```

  发现数据库编码已改成utf8

8. 创建自己flask项目里的数据库

   ```
   CREATE DATABASE rlxiot
   ```

   到此，mysql就安装完成啦！

上传项目文件

1. 用scp命令或其他工具将自己的项目文件上传到服务器，上传前记得先生成requirements.txt文件，这里我上传到root目录下

   ```
   scp -r rlxiot root@47.93.15.120:/root/
   ```

   上面项目文件名，用户名以及服务器ip换成自己的即可

2. 安装项目引用的包

   ```
   pip install -r requirements.txt
   ```

安装Gunicorn

1. 使用pip安装Gunicorn

   ```
   pip install Gunicorn
   ```

2. 使用gunicorn启动项目

   ```
   gunicorn --workers=3 manage:app -b 0.0.0.0:8080
   ```

   上面的manage和app分别填自己的启动文件和项目应用

   如果服务器8080端口开放，浏览器输入ip:8080就可以通过看到到我们的项目

安装Nginx

1. yum安装Nginx

   ```
   yum install nginx
   ```

2. 启动Nginx

   ```
   nginx
   ```

访问ip，就可以看到Nginx欢迎的界面了

3. 配置Nginx

使用vim打开/etc/nginx/conf.d/default.conf文件

```
vim /etc/nginx/nginx.conf
```

将http中的server替换为以下内容

```
server {
    listen      80;

    # 这里的 your.domain.com 换成你购买的域名
    server_name _;

    location / {
        # 转发请求给 Gunicorn
        proxy_pass      http://127.0.0.1:8080;
        proxy_redirect  off;

        # 为了能正常运行，重写请求头
        proxy_set_header    Host                $host:80;
        proxy_set_header    X-Real-IP           $remote_addr;
        proxy_set_header    X-Forwarded-For     $proxy_add_x_forwarded_for;
    }
}
```

4. 重启nginx

```
nginx -s reload
```

如果没有报错，应该配置好了，这个时候启动gunicorn，就可以通过ip直接访问了。

安装Supervisor

1. 使用pip安装supervisor

```
pip install supervisor
```

2. 创建supervisor配置文件

```
vim supervisor.conf
```

加入以下内容并保存

```
[program:app]
command=gunicorn -w4 manage:app -b 0.0.0.0:8080

directory=/root/rlxiot

autostart=true
sutorestart=true
stopsignal=QUIT
stopsgroup=true
killsgroup=true

user=root
redirect_stderr=true
stdout_logfile=/root/rlxiot/log/supervisor.log

[supervisord]
environment=LC_ALL='en_US.UTF-8',LANG='en_US.UTF-8'
```

3. 启动supervisor并重启nginx

```
supervisord supervisor.conf
nginx -s reload
```

然后就可以在浏览器输入ip访问项目，程序挂了也可以自动重启e。