**Запуск приложений flask**

**с использованием веб-сервера NGINX**

1. Обновите пакеты

sudo apt-get update

sudo apt upgrade

1. Установка необходимого программного обеспечения

sudo apt-get install nginx

sudo apt-get install python3-pip

sudo pip install flask

sudo apt-get install build-essential python-dev

sudo pip3 install flask uwsgi

uWSGI, программное приложение, в котором размещаются службы. build-essentials и python-dev (-компоненты для обслуживания служб)

1. Создать приложение Flask

создадим папку в домашнем каталоге («/home/oleksandr»), но сначала убедимся, что мы находимся в каталоге /home/oleksandr, используя:

cd ~

После этого создадим экземпляр приложения

mkdir flasktesting

после изменим настройку папки так, чтобы она имела признаки web-приложения

sudo chown www-data /home/oleksandr/flasktesting

войдем в эту папку

cd flasktesting

создадим новый файл с именем «fossapp.py» в нашем каталоге «/home/oleksandr/flasktesting».

sudo vi /home/pi/fossapp.py

вставим тект

from flask import Flask  
app = Flask(\_\_name\_\_)  
  
@app.route("/")  
def index():  
 return "<html><body><h1>Test site running under Flask</h1></body></html>"  
if \_\_name\_\_ == "\_\_main\_\_":  
 app.run(host='0.0.0.0',debug=True)

в редакторе vi сохранение выглядит как :w а выход :q

проверим наш IP-адрес перед тестированием приложения с помощью следующей команды

ifconfig

увидим ip (192.168.0.5)

наберем IP-адрес распберри в браузере

http://192.168.0.5

(должна отобразиться страница приветствия NGINX по умолчанию)

1. Тестирование uWSGI

uwsgi --socket 0.0.0.0:8000 --protocol=http -w fossapp:app

Это разрешит зайти через порт 8000 на IP-адрес Распберри.

http://192.168.0.5:8000

должны увидеть такую ​​страницу:

Test site running under Flask

Чтобы вернуться в терминал, используйте CTRL-C

1. Создаем файл инициализации uWSGI

sudo nano uwsgi.ini

пишем в него:

[uwsgi]

chdir = /home/oleksandr/flasktesting  
module = fossapp:app  
master = true  
processes = 1  
threads = 2  
uid = www-data  
gid = www-data  
socket = /tmp/flasktesting.sock  
chmod-socket = 664  
vacuum = true  
die-on-term = true

сохранение: CTRL-W, ENTER, CTRL-X

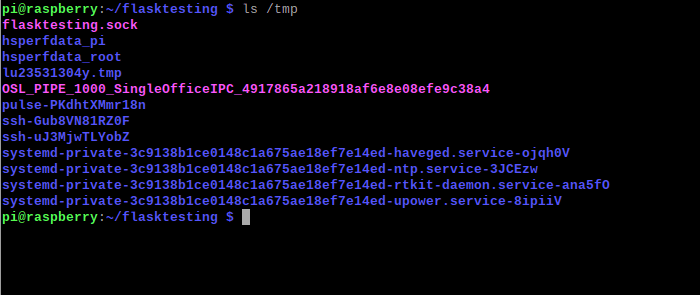
файл содержит ссылки на каталог проекта (/home/oleksandr/flasktesting) и файл python flask «fossapp.py». В этой части файла «flasktesting.sock» на данный момент не существует. Однако файл будет автоматически создан, когда это потребуется для uWSGI

1. Тестируем файл инициализации uWSGI.

uwsgi --ini uwsgi.ini

используя **второе** соединение SSH, проверяем успешность создания файла сокета с помощью следующей команды:

ls /tmp



должны увидеть «flasktesting.sock» среди файлов (этот файл появляется только во время работы uWSGI)

Используйте CTRL-C, чтобы остановить запуск uWSGI

1. Настройка Nginx для использования uWSGI

Нужно настроить NGINX для направления трафика на uWSGI. Обычно это называют «обратным прокси».

а) первый шаг — удалить сайт по умолчанию:

sudo rm /etc/nginx/sites-enabled/default

После этого создать новый файл с именем «flasktesting\_proxy»

sudo nano /etc/nginx/sites-available/flasktesting\_proxy

скопировать следующее в файл flasktesting\_proxy:

server {  
listen 80;  
server\_name localhost;  
  
location / { try\_files $uri @app; }  
  
location @app {  
include uwsgi\_params;  
uwsgi\_pass unix:/tmp/flasktesting.sock;  
}  
  
}

CTRL-X, CTRL-Y, and ENTER

Для завершения создать символическую ссылку из этого файла в каталог «sites-enabled», используя:

sudo ln -s /etc/nginx/sites-available/flasktesting\_proxy /etc/nginx/sites-enabled

1. Перезапустить Nginx

sudo systemctl restart nginx

Примечание. Если зайти на свой IP-адрес после перезапуска NGINX, вы должны получить сообщение об ошибке «502 Bad Gateway». Это должно быть ожидаемым результатом, потому что uWSGI в настоящее время не работает, а NGINX пытается передать ему запрос браузера, отсюда и ошибка

1. Запуск uWSGI при загрузке pi

uWSGI должен запускаться автоматически каждый раз при перезапуске Pi. Чтобы добиться автоматического перезапуска, мы будем использовать службу systemd.

Для этого сначала перейдите в системный каталог:

cd /etc/systemd/system

После этого создайте юнит-файл для uWSGI

sudo nano uwsgi.service

и вставьте следующее:

[Unit]  
Description=uWSGI Service  
After=network.target  
  
[Service]  
User=www-data  
Group=www-data  
WorkingDirectory=/home/pi/fosstest/  
ExecStart=/usr/local/bin/uwsgi --ini /home/pi/fosstest/uwsgi.ini  
  
[Install]  
WantedBy=multi-user.target

CTRL-X, CTRL-Y, and ENTER

Чтобы демон подхватил новый юнит, мы перезапустим его с помощью следующей команды:

sudo systemctl daemon-reload

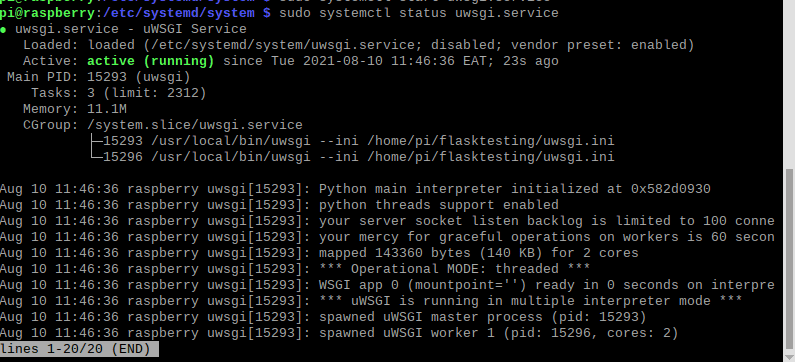
Теперь запускаем службу

sudo systemctl start uwsgi.service

Чтобы завершить, давайте запустим проверку состояния службы:

sudo systemctl status uwsgi.service

После запуска команды проверки состояния вы должны увидеть такой экран:



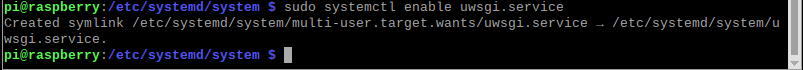
самым критическим участком является строка, определяющая «активно» («работает»). Это показывает, что служба запущена и работает. Чтобы вернуться в командную строку из службы, нажмите CTRL-C.

чтобы служба запускалась автоматически при каждой перезагрузке:

sudo systemctl enable uwsgi.service

После запуска команды вы получите такой вывод:

“Created symlink /etc/systemd/system/multi-user.target.wants/uwsgi.service /etc/systemd/system/uwsgi.service”



1. На этом этапе вы можете перезапустить Raspberry Pi, используя:

sudo reboot

После этого на своем ПК посетите IP-адрес Pi, как мы делали это раньше. Теперь вы должны автоматически увидеть страницу приложения Flask в браузере из-за настроенного нами автоматического перезапуска.

К этому моменту вы должны быть довольны собой, так как теперь у вас есть фляжное приложение, обслуживаемое NGINX.

1. Что касается перезагрузки (необязательно)

Этот шаг является необязательным. С текущими изменениями настройки файла «fossapp.py» файл не будет отображаться, несмотря на обновление браузера. Он может отображаться только при перезапуске службы uWSGI.

Вы можете заставить uWSGI автоматически подхватівать изменения, внесенные в файл, добавив директиву «touch-reload» в файл uwsgi.ini.

Используйте следующую команду для редактирования файла:

sudo nano uwsgi.ini

допишем директиву в новой строке:

touch-reload = /home/oleksandr/flasktesting/fossapp.py

CTRL-X, CTRL-Y, and ENTER

sudo reboot

После загрузки Pi все внесенные изменения будут видны в браузере при обновлении страницы.