Python Programming

Apache2 Python Flask Application

Dr Diarmuid Ó Briain

Version: 1.0

Ollscoil Teicneolaíochta an Oirdheiscirt
South East Technological University

Table of Contents

Illustration Index

This page is intentionally blank

1 Introduction

Flask is a popular web development framework written in Python. Flask is considered a micro-framework because it does not require particular tools or libraries. Flask can be installed and deployed with Apache and Web Server Gateway Interface (WSGI).

2 Updating the system

Update our system to make sure that all the installed packages are up to date.

```
~$ sudo apt -y update && sudo apt -y upgrade
```

3 Install Flask

Install Flask.

```
~$ python3 -m pip install Flask
```

4 Setup Flask

4.1 Create a flask directory

Create a directory for the Flask files.

```
~$ mkdir ~/web
```

4.2 Test app

Create a test app.

```
~$ cat <<EOM > ~/web/init.py
from flask import Flask, render_template

app = Flask(__name__)

@app.route("/")
def home():
    return render_template("index.html")

if __name__ == "__main__":
    app.run(debug=True)
EOM
```

4.3 Template

Create a templates directory and an **index.html** file within it.

```
~$ mkdir ~/web/templates
~$ cat <<EOM > web/templates/index.html
<html>
  <head>
  </head>
  <body>

<center> <h1>Hello World, Flask</h1> </center>

  </body>
</html>
EOM
```

4.4 Test the app

Test the app.

```
~$ python3 ~/web/init.py
 * Serving Flask app 'init'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production
deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
 * Restarting with stat
 * Debugger is active!
```

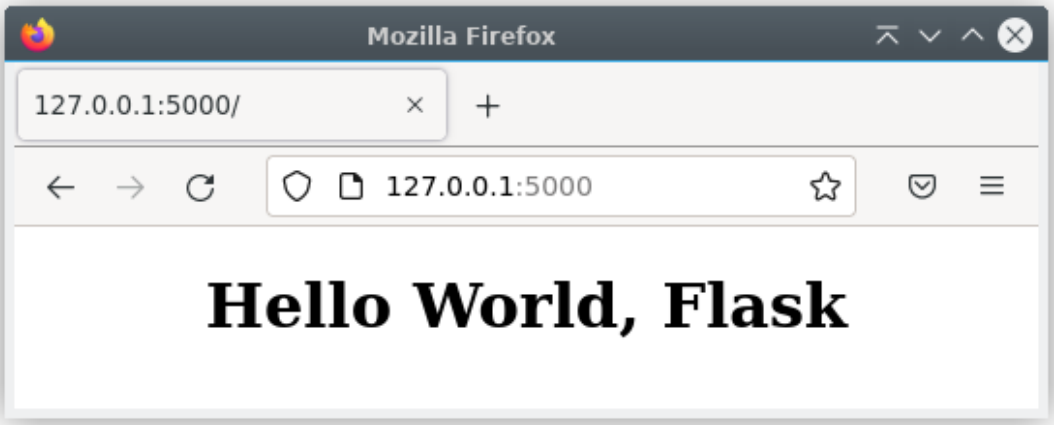

Figure 1: Browse to the webserver at the localhost

Note the line that appear in the test server.

```
127.0.0.1 - - [14/Sep/2022 23:20:53] "GET / HTTP/1.1" 200 -
```

5 Apache2 Webserver

5.1 Installing Apache2

Install Apache2 webserver and move the directory, web, to its root, **/var/www/html**.

```
~$ sudo apt install apache2
```

5.2 Create the app wsgi file

Create a Web Server Gateway Interface (WSGI) file which adds the Python virtual environment packages and the website app paths and then launches the Flask app **app = Flask(__name__)** from **init.py**.

```
~$ cat <<EOM > ~/web/app.wsgi
import sys

sys.path.insert(0, "/home/debian/.venv/lib/python3.9/site-packages")
sys.path.insert(0, "/var/www/html/web")

from init import app as application
EOM
```

5.3 Move the webapp to the Apache2 Webserver

Now move the web app under the Apache2 server root.

```
~$ sudo mv ~/web /var/www/html/
~$ sudo chown -R www-data: /var/www/html/web
```

5.4 Apache2 site configuration file

Create a site configuration file by replacing the default, this points to the WSGI and the directory for the app.

```
~$ cd /etc/apache2/sites-available/
/etc/apache2/sites-available$ sudo mv 000-default.conf 000-default.conf.orig
/etc/apache2/sites-available$ cat <<EOM | sudo tee 000-default.conf
<VirtualHost *:80>
        ServerAdmin webmaster@localhost

        WSGIScriptAlias / /var/www/html/web/app.wsgi
        <Directory /var/www/html/web>
            Order allow,deny
            Allow from all
        </Directory>

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined

</VirtualHost>
EOM
```

5.5 Install the WSGI and enable it for Apache2

```
/etc/apache2/sites-available$ cd ~
~$ sudo apt install libapache2-mod-wsgi-py3
~$ sudo a2enmod wsgi
Enabling module wsgi.
```

5.6 Launch Apache2 Server

```
~$ sudo systemctl restart apache2

~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; enabled;
vendor preset: enabled)
     Active: active (running) since Wed 2022-09-14 23:09:13 IST; 29s
ago
       Docs: https://httpd.apache.org/docs/2.4/
    Process: 3751 ExecStart=/usr/sbin/apachectl start (code=exited,
status=0/SUCCESS)
   Main PID: 3755 (apache2)
      Tasks: 55 (limit: 9493)
     Memory: 23.5M
        CPU: 66ms
     CGroup: /system.slice/apache2.service
             ├─3755 /usr/sbin/apache2 -k start
             ├─3756 /usr/sbin/apache2 -k start
             └─3757 /usr/sbin/apache2 -k start

Sep 14 23:09:13 debian systemd[1]: Starting The Apache HTTP Server...
Sep 14 23:09:13 debian systemd[1]: Started The Apache HTTP Server.
```

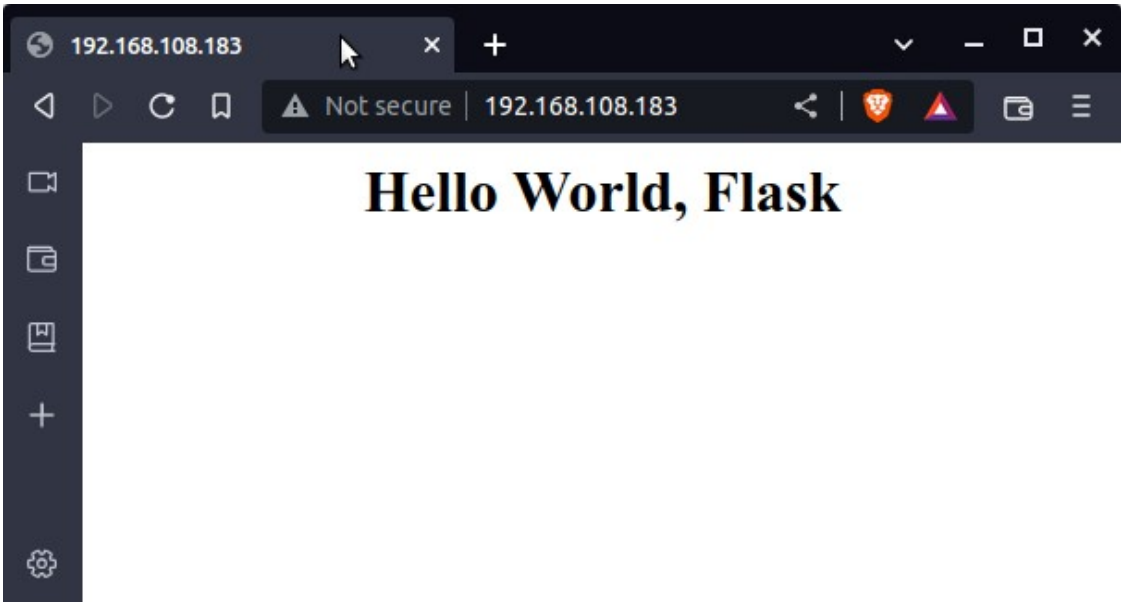

Figure 2: Test the webserver from a remote host

5.7 Tree in the Apache2 root directory

```
~$ tree /var/www/html/web
/var/www/html/web
├── app.wsgi
├── init.py
├── __pycache__
│   └── init.cpython-39.pyc
└── templates
    └── index.html

2 directories, 4 files
```

6 Securing the website

This section only applies if your server is on the Internet and you have a genuine domain for your server. For local installation skip section 5.

6.1 Install an SSL Certificate

Install **cerebot** and the Apache plugin.

```
~$ sudo apt install -y certbot python3-certbot-apache
~$ certbot --version
certbot 1.12.0
```

Confirm the plugin for Apache is available.

```
~$ certbot plugins

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
* apache
Description: Apache Web Server plugin
Interfaces: IAuthenticator, IInstaller, IPlugin
Entry point: apache = certbot_apache._internal.entrypoint:ENTRYPOINT
```

Install the certificate and configure the Apache HTTP Server.

```
~$ sudo certbot --apache
Enter email address (Enter 'c' to cancel): ada@lovelace.com
You must agree to register with the ACME server. (Y)es/(N)o: Y
Let the EFF send you an email (Y)es/(N)o: Y
Enter in your domain name(s): www.lovelace.com
```

6.2 Update the firewall

Update the firewall to permit access to the Hypertext Transfer Protocol Secure (HTTPS) port 443.

```
~$ sudo ufw allow 443/tcp
Rule added
Rule added (v6)
~$ sudo ufw reload
Firewall reloaded

~$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
1194/udp                   ALLOW       Anywhere
80                         ALLOW       Anywhere
22/tcp                     ALLOW       Anywhere
80/tcp                     ALLOW       Anywhere
443/tcp                    ALLOW       Anywhere
1194/udp (v6)              ALLOW       Anywhere (v6)
80 (v6)                    ALLOW       Anywhere (v6)
22/tcp (v6)                ALLOW       Anywhere (v6)
80/tcp (v6)                ALLOW       Anywhere (v6)
443/tcp (v6)               ALLOW       Anywhere (v6)
```

6.3 Annual certificate update

When the certificate runs out, update by executing this command:

```
~$ sudo certbot certonly --apache
```

This page is intentionally blank