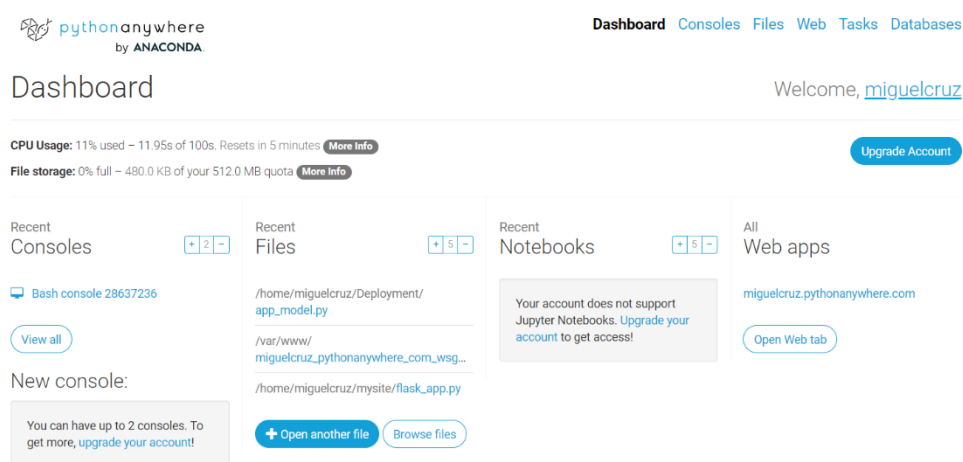


Pythonanywhere

Vamos a desplegar una api en el servidor de pythonanywhere, estos son los pasos a seguir:

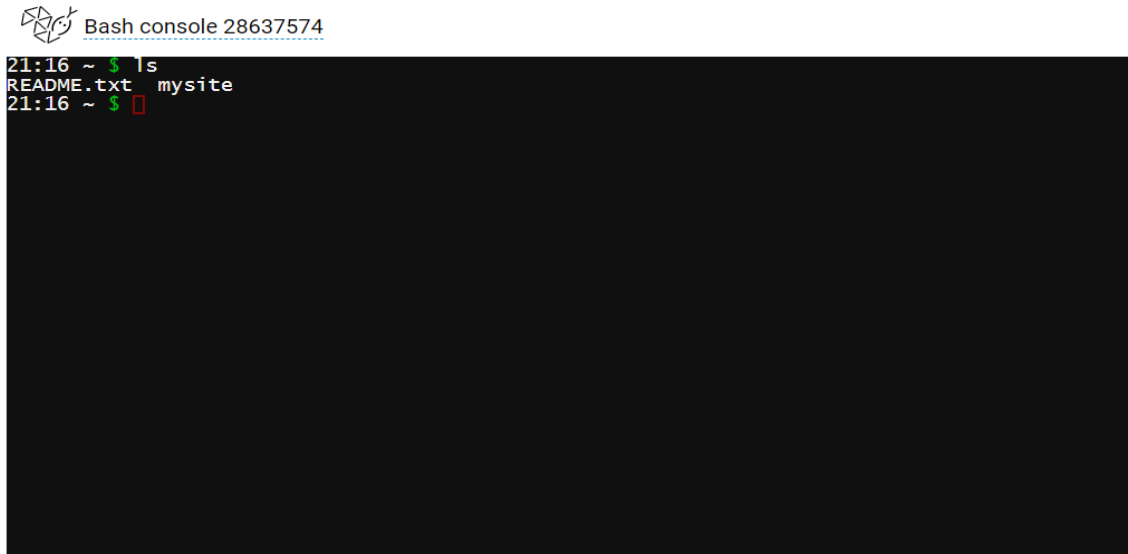
1. Entramos en [pythonanywhere](https://pythonanywhere.com) y creamos una cuenta, nos debe aparecer la siguiente ventana luego de crear la cuenta:



2. Vamos a la sección de “New console” para crear la consola o terminal desde donde vamos a manejar los archivos de nuestra API:



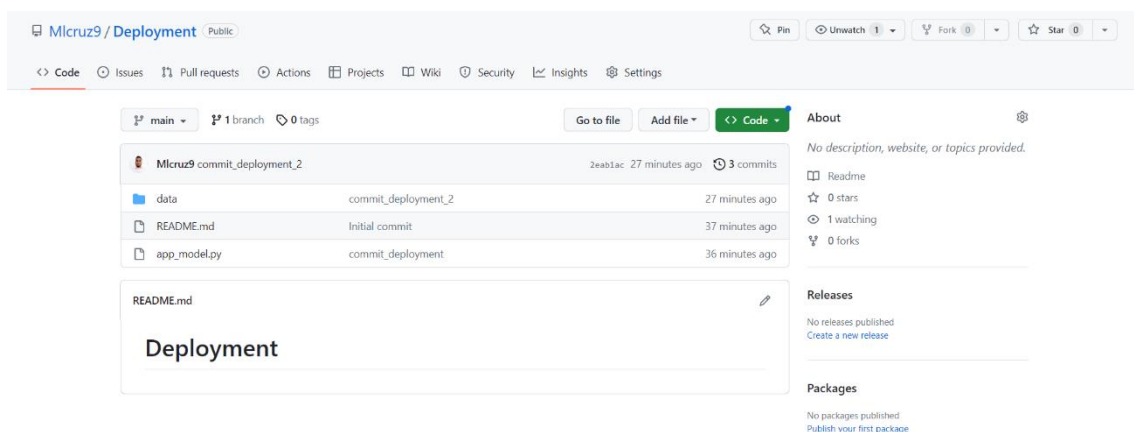
3. Clicamos en “Bash” y nos aparecerá la siguiente ventana:



```
Bash console 28637574
21:16 ~ $ ls
README.txt  mysite
21:16 ~ $
```

Si escribimos “ls” y presionamos “enter” obtendremos la lista de archivos y directorios que tenemos en el servidor, en este caso solo un archivo “README” y una carpeta “mysite” creados por defecto.

4. Creamos un repositorio en Git Hub en donde cargaremos la api y los archivos que necesita la misma para ejecutarse:



En mi caso, mi repositorio se llamará “Deployment” y dentro de el estarán cargados los archivos de los que tirará mi api, `app_model.py` (la api) y `data` (carpeta en donde se encuentra el modelo predictivo que queremos desplegar).

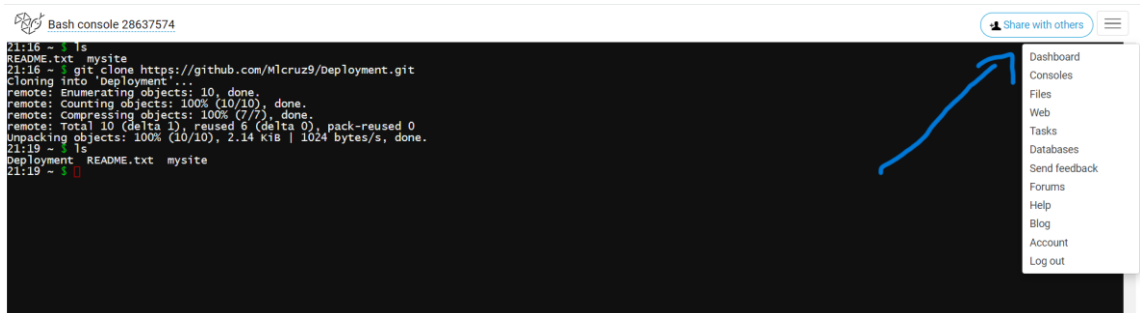
5. Vamos a nuestra consola “bash” y nos clonamos el repositorio:



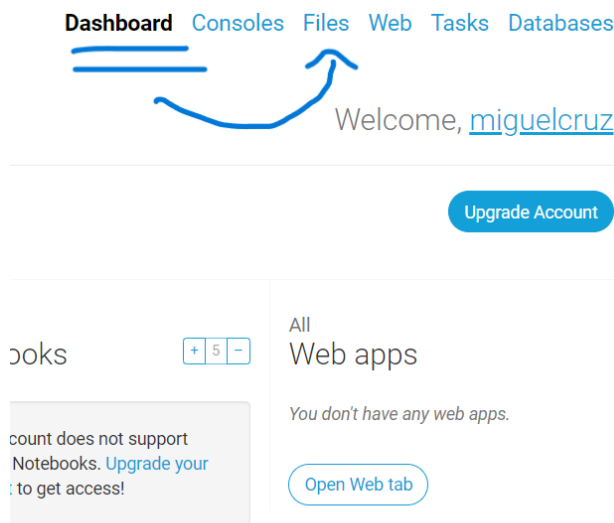
```
21:16 ~ $ ls
README.txt  mysite
21:16 ~ $ git clone https://github.com/Mlcruz9/Deployment.git
Cloning into 'Deployment'...
remote: Enumerating objects: 10, done.
remote: Counting objects: 100% (10/10), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 10 (delta 1), reused 6 (delta 0), pack-reused 0
Unpacking objects: 100% (10/10), 2.14 KiB | 1024 bytes/s, done.
21:19 ~ $ ls
Deployment  README.txt  mysite
21:19 ~ $
```

Si hacemos un “ls” nuevamente podremos observar que tenemos una nueva carpeta, la carpeta con los archivos del repositorio, llamada “Deployment”.

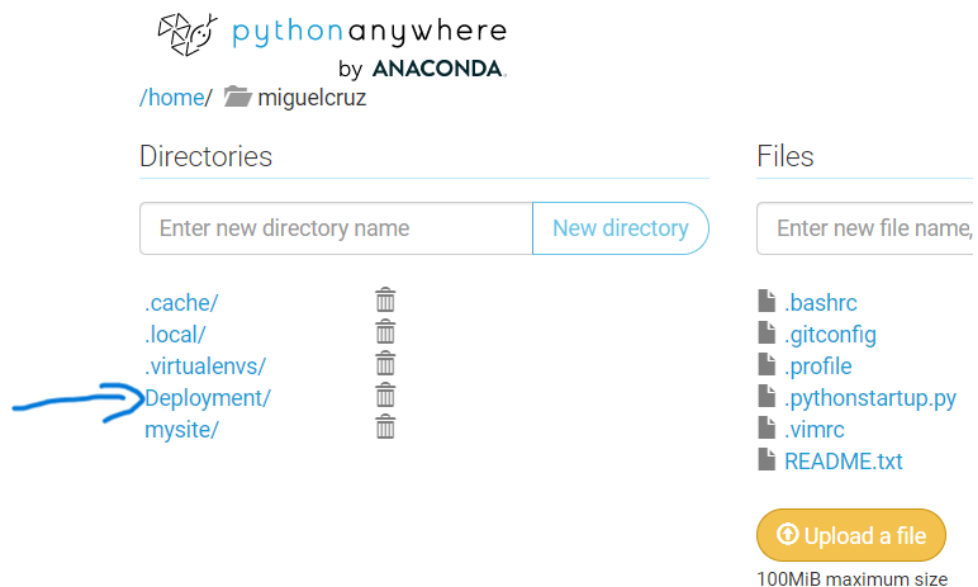
6. En la esquina superior derecha aparece un símbolo con tres líneas horizontales, cliqueamos en el y vamos a “dashboard”:



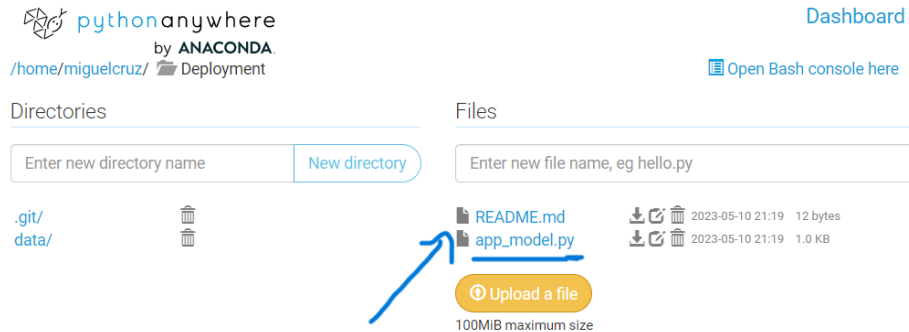
7. Nos encontramos en “dashboard”, ahora vamos a “files”:



8. En files podremos ver los directorios que tenemos en nuestro servidor, también veremos nuestra carpeta “Deployment”, proveniente del repositorio:



9. Entramos en la carpeta deployment, y luego en el archivo de nuestra api “app_model.py”:



Archivo app_model.py:

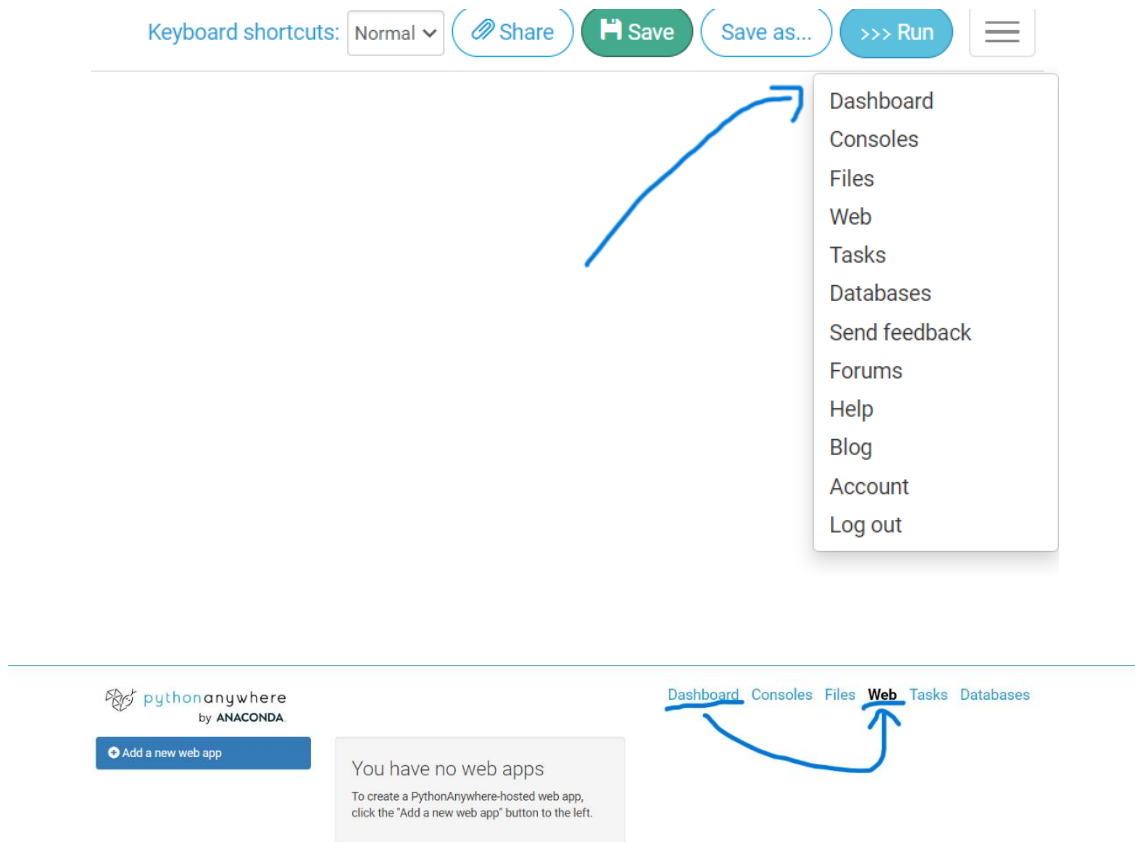
```
1 from flask import Flask, request, jsonify
2 import os
3 import pickle
4 from sklearn.model_selection import cross_val_score
5 import pandas as pd
6
7
8 os.chdir(os.path.dirname(__file__))
9
10 app = Flask(__name__)
11 app.config['DEBUG'] = True
12
13 @app.route("/", methods=['GET'])
14 def hello():
15     return "Bienvenido a mi API del modelo advertising"
16
17 # 1. Endpoint que devuelve la predicción de los nuevos datos enviados mediante argumentos en la llamada
18 @app.route('/v1/predict', methods=['GET'])
19 def predict():
20     model = pickle.load(open('data/advertising_model', 'rb'))
21
22     tv = request.args.get('tv', None)
23     radio = request.args.get('radio', None)
24     newspaper = request.args.get('newspaper', None)
25
26     if tv is None or radio is None or newspaper is None:
27         return "Missing args, the input values are needed to predict"
28     else:
29         prediction = model.predict([[tv, radio, newspaper]])
30         return "The prediction of sales investing that amount of money in TV, radio and newspaper is: " + str(round(prediction[0], 2)) + 'k €'
31
32 app.run()
```

10. Al final del archivo dejamos comentado el “app.run()” porque no lo vamos a utilizar:

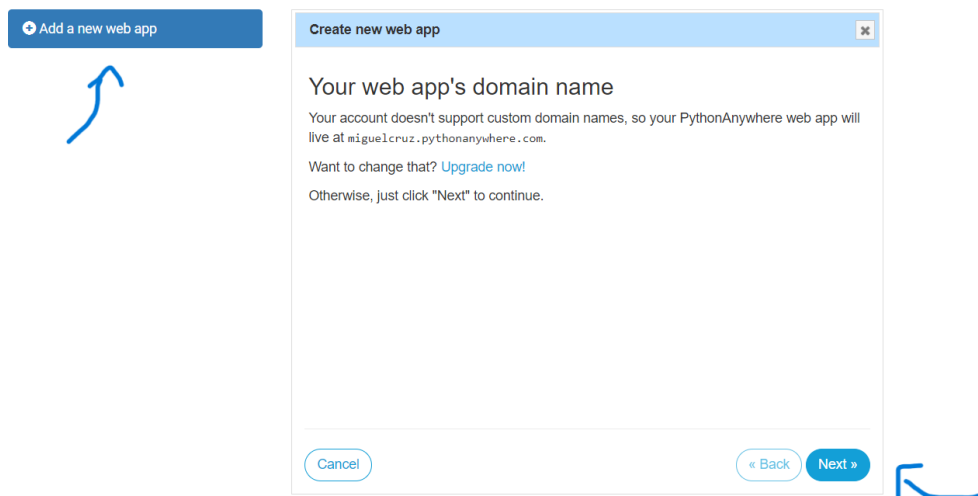
```
31
32 # app.run()
```

11. Vamos de nuevo a “dashboard” y luego a “web”:

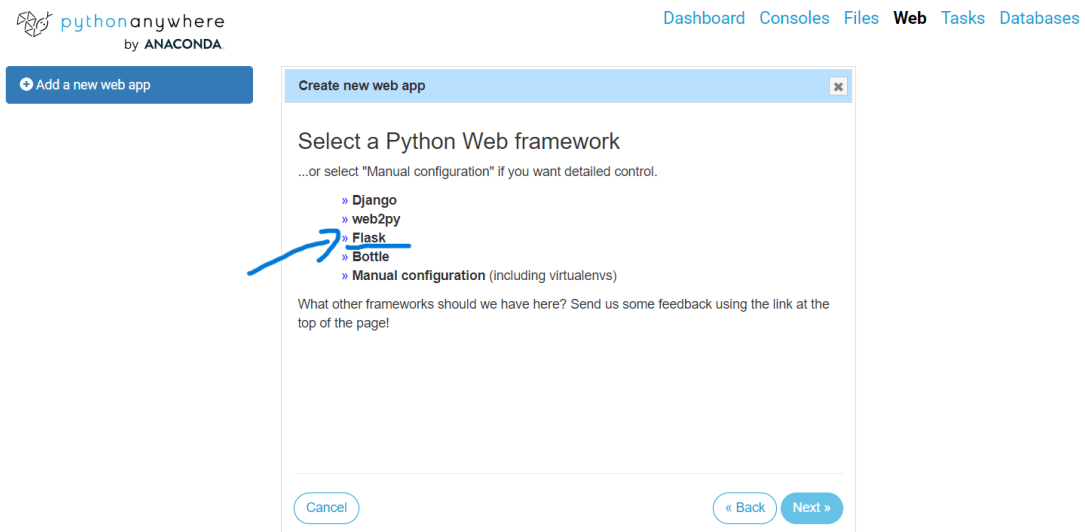
Dashboard:



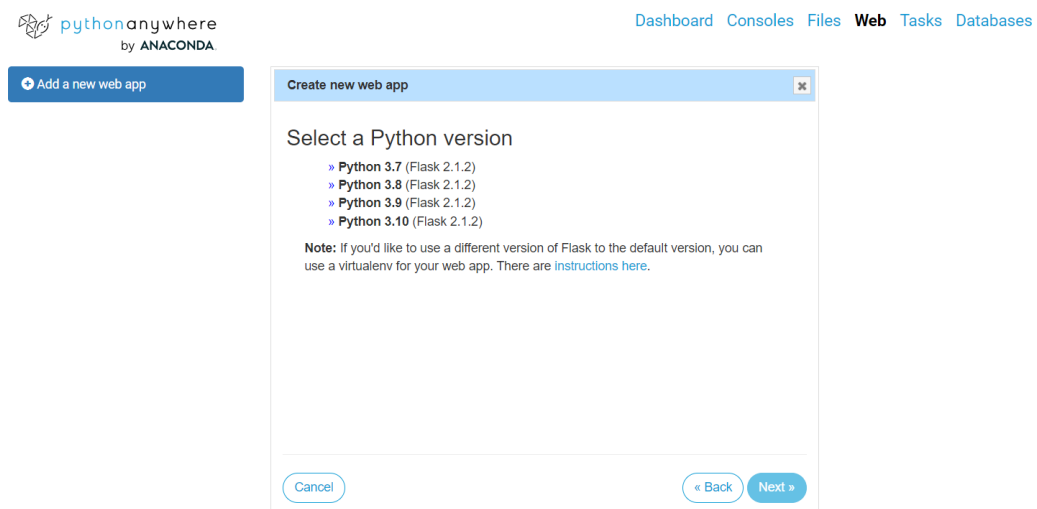
12. Cliqueamos en “add a new web app” y luego “next”:



13. Nos aparecen varios frameworks, nosotros utilizaremos flask y cliqueamos “next”:



14. Selecciona tu versión de Python. En mi caso será 3.7:



15. Luego la página crea una api por defecto en el directorio que indica (“/mysite/flask_app.py”). Cliqueamos a “next”:

Create new web app

Quickstart new Flask project

Enter a path for a Python file you wish to use to hold your Flask app. If this file already exists, its contents will be overwritten with the new app.


Path

Cancel

« Back

Next »

16. Vamos nuevamente a “Web”:

pythonanywhere
by ANACONDA

Dashboard Consoles Files **Web** Tasks Databases

All done! Your web app is now set up. Details below.

miguelcruz.pythonanywhere.com

⚙ Add a new web app

Configuration for miguelcruz.pythonanywhere.com

Reload:

Reload miguelcruz.pythonanywhere.com

Best before date:

We're happy to host your free website – and keep it free – for as long as you want to keep it running, but you'll need to log in at least once every three months and click the "Run until 3 months from today" button below. We'll send you an email a week before the site is disabled so that you don't forget to do that. [See here for more details.](#)

This site will be disabled on **Thursday 10 August 2023**


Run until 3 months from today

Paying users' sites stay up forever without any need to log in to keep them running.

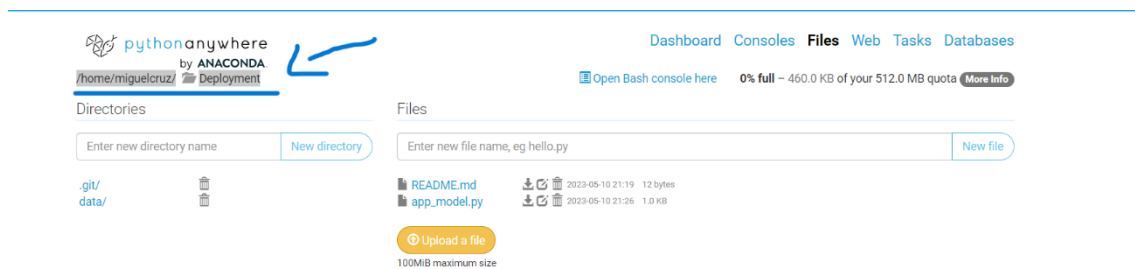
17. Bajamos a la sección de “Code” como se indica a continuación:

Code:

What your site is running.

Source code:	/home/miguelcruz/mysite	Go to directory
Working directory:	/home/miguelcruz/	Go to directory
WSGI configuration file:	/var/www/miguelcruz_pythonanywhere_com_wsgi.py	
Python version:	3.7 	

18. Donde marca “Source code”, cambiaremos el directorio por el directorio en donde habíamos alojado nuestra api (donde clonamos nuestro repositorio), en mi caso “/home/miguelcruz/Deployment”. Si queremos ver la dirección donde habíamos alojado la info de nuestro repositorio podemos ir a “files”:



Cambiando el directorio:

Code:

What your site is running.

Source code:	/home/miguelcruz/Deployment	Go to directory
Working directory:	/home/miguelcruz/	Go to directory
WSGI configuration file:	/var/www/miguelcruz_pythonanywhere_com_wsgi.py	
Python version:	3.7 	

19. Cliqueamos en el hipervínculo a la derecha de WSGI configuration file
("/var/www/miguelcruz_pythonanywhere_com_wsgi.py" en mi caso):



/var/www/miguelcruz_pythonanywhere_com_wsgi.py

```
1 # This file contains the WSGI configuration required to serve up your
2 # web application at http://<your-username>.pythonanywhere.com/
3 # It works by setting the variable 'application' to a WSGI handler of some
4 # description.
5 #
6 # The below has been auto-generated for your Flask project
7
8 import sys
9
10 # add your project directory to the sys.path
11 project_home = '/home/miguelcruz/mysite'
12 if project_home not in sys.path:
13     sys.path = [project_home] + sys.path
14
15 # import flask app but need to call it "application" for WSGI to work
16 from flask_app import app as application # noqa
17
```

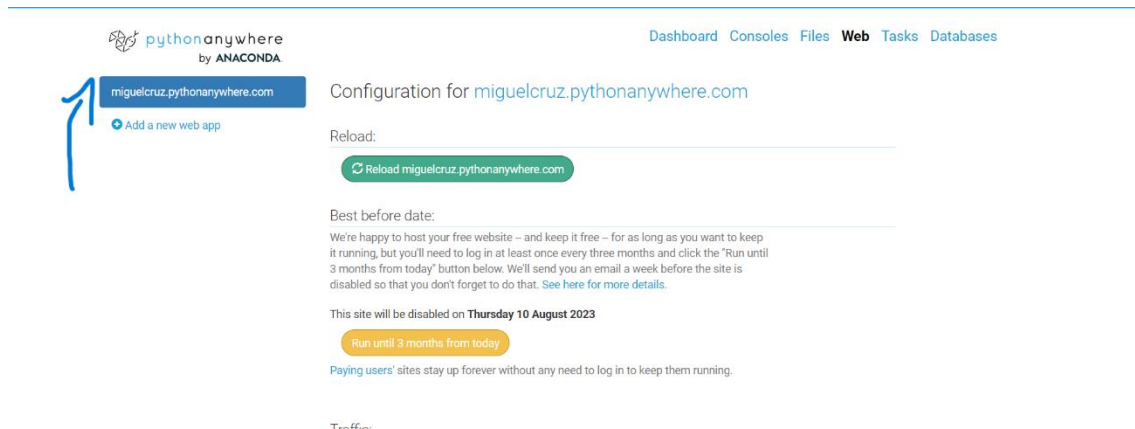
20. Arriba "flask_app" está subrayado, es el nombre de la api que crea por defecto
pythonanywhere, lo sustituimos por el nombre de nuestra api ("app_model"):



/var/www/miguelcruz_pythonanywhere_com_wsgi.py (unsaved changes)

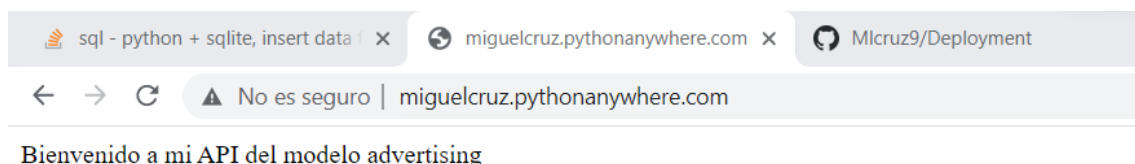
```
1 # This file contains the WSGI configuration required to serve up your
2 # web application at http://<your-username>.pythonanywhere.com/
3 # It works by setting the variable 'application' to a WSGI handler of some
4 # description.
5 #
6 # The below has been auto-generated for your Flask project
7
8 import sys
9
10 # add your project directory to the sys.path
11 project_home = '/home/miguelcruz/Deployment'
12 if project_home not in sys.path:
13     sys.path = [project_home] + sys.path
14
15 # import flask app but need to call it "application" for WSGI to work
16 from app_model import app as application # noqa
17
```

21. Vamos de nuevo a web y cliqueamos en el rectángulo azul:



22. Si está todo bien, debería desplegar ya nuestra api, a la que podremos agregar argumentos para realizar predicciones:

Página inicial:



Ejemplo con argumentos:

