



Deployment

Code example:

```
# file: app.py

from flask import Flask, jsonify
from flask.globals import request
from products import products

app = Flask(__name__)

@app.route('/')
def index():
    return jsonify({"message": "cristian"})

@app.route('/user/<string:userid>', methods=['GET'])
def get_user(userid):
    if request.method == 'GET':
        pass
    return 'resultado %s' % userid

if __name__ == '__main__':
    app.run()
```

Requerimientos:

```
pip install gunicorn
```

Creamos nuestro entorno virtual con:

Cambiar el archivo Pipfile.lock y Pipfile a la version de python funcional en heroku
actual:3.8

```
pipenv install flask gunicorn flask_cors ply
```

Ingresamos a nuestro entorno virtual

```
pipenv shell
```

creamos el file "Procfile". sin extension.

```
web: gunicorn app:app
```

generamos el archivo requirements.txt

```
pip freeze > requirements.txt
```

Nota: Si al subir se genera un error el archivo de que no esta actualizado el archivo pickfile.lock ingresamos el siguiente comando.

```
pipenv lock
```

by Heroku

Create a new Git repository

Initialize a git repository in a new or existing directory

```
$ cd my-project/  
$ git init  
$ heroku git:remote -a app-jolc
```

Deploy your application

Commit your code to the repository and deploy it to Heroku using Git.

```
$ git add .  
$ git commit -am "make it better"  
$ git push heroku master
```



You can now change your main deploy branch from "master" to "main" for both manual and automatic deploys, please follow the instructions [here](#).