

# WORKSHOP “DEPLOY FLASK CON HEROKU”

Clara Piniella - TA Data 2020

## MENSAJE PREVIO AL TALLER

¡Buenas @channel! Mañana a lo largo del día, aún no sé deciros hora, realizaremos un workshop práctico de cómo subir o deployar una API a Heroku. Si bien yo os proveeré de la API que usaremos de ejemplo y os iré guiando en los pasos, de forma que os sea fácil de entender, necesito que tengáis instalado tanto **Heroku CLI** como **Docker** en vuestro ordenador. Tenemos una semana de traca final, como no podía ser de otro modo, por lo que os agradecería que lo tengáis instalado mañana. Solo instalado, no hace falta que hagáis nada más. ¡Gracias!

- Para Heroku CLI, si eres de Windows el de 64 bits, si eres de mac solo hay una opción: <https://devcenter.heroku.com/articles/heroku-cli>
- Para Docker la cuenta FREE e instalaos Docker Desktop, en ambos casos elegir Stable. <https://www.docker.com/products/docker-desktop>

1. Fork [https://github.com/Clapiniella/destination\\_Heroku](https://github.com/Clapiniella/destination_Heroku)
2. Clona a tu ordenador tu repo forkeado:  

```
`git clone https://github.com/TUUSUARIO/destination_Heroku`
```
3. Abre el repo con Visual Studio Code.

Archivos BÁSICOS y MÍNIMOS que no podrán ir dentro de ninguna carpeta:

- Dockerfile
- requirements.txt
- el archivo main.py o api.py que será lo que ejecute tu servidor Flask

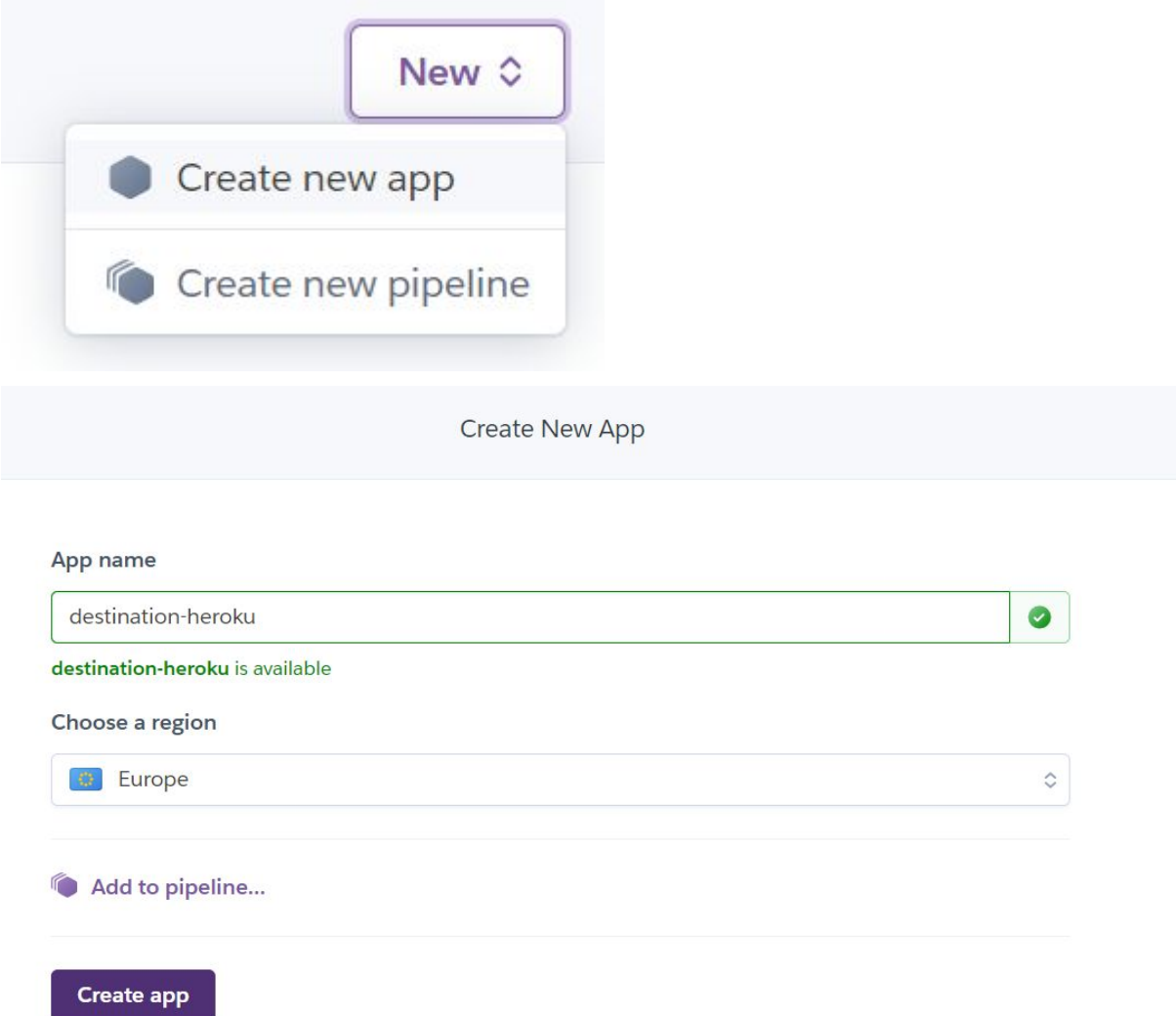
Dentro de este punto es muy importante la configuración del puerto, en este caso deberá especificarse como `os.getenv("PORT", unnumerodepuertoenlocal)`. Este sería un ejemplo:

```
app.run(host='0.0.0.0',port=os.getenv("PORT", 5000), debug=True)
```

Antes de subir la API a Heroku, asegúrate de que tu repo tiene estos archivos, y que por supuesto, la API funciona en local.

## DEPLOY WITH HEROKU

1. Crea una cuenta en Heroku.  
<https://signup.heroku.com/login>
2. Una vez tengas la cuenta, accede al dashboard de Heroku y crea una APP.  
<https://dashboard.heroku.com/apps>



New ↕

- Create new app
- Create new pipeline

Create New App

App name

destination-heroku ✓

destination-heroku is available

Choose a region

Europe ↕

Add to pipeline...

Create app

Como yo he elegido el nombre destination-heroku, ya no estará disponible por lo que os recomiendo que o bien le deis un nombre vosotros o añadáis los dos primeros números de vuestro DNI al final, ejemplo: *destination-heroku76*.

3. Ahora abre tu repo en git bash/ terminal y escribe el siguiente comando:  
**heroku login**  
Pulsa Enter y se abrirá una ventana para que metas tus credenciales de Heroku.
4. Enlaza tu repo de Github a la app de Heroku con el siguiente comando. Es importante que tengas permisos de push.  
**heroku git:remote -a destination-heroku76**

```
clara@TA-Data-Science MINGW64 ~/Desktop/The_Bridge/destination_Heroku (main)
$ heroku git:remote -a destination-heroku
set git remote heroku to https://git.heroku.com/destination-heroku.git

clara@TA-Data-Science MINGW64 ~/Desktop/The_Bridge/destination_Heroku (main)
$ git remote -v
heroku https://git.heroku.com/destination-heroku.git (fetch)
heroku https://git.heroku.com/destination-heroku.git (push)
origin https://github.com/Clapiniella/destination_Heroku.git (fetch)
origin https://github.com/Clapiniella/destination_Heroku.git (push)
```

5. Los siguientes comandos que deberás ejecutar son los siguientes:

```
heroku container:login
```

```
heroku container:push web
```

puede que este paso tarde unos minutos, ten paciencia

```
heroku container:release web
```

Sabrás que tu API se ha subido correctamente si ves esto: **Releasing images web to destination-heroku... done**

Este es el objetivo, cambia el nombre de la app por el tuyo:

<https://destination-heroku.herokuapp.com/>

## MÁS INFO

<https://codigofacilito.com/articulos/deploy-flask-heroku>

<https://devcenter.heroku.com/articles/getting-started-with-python>