

SmartData Construcción

Documentación

Deployment

SmartData Construcción

Documento producido por PostData

Agosto del 2019

CLHM

Introducción

Este documento tiene por objetivo, realizar de manera simple el despliegue (deploy) de una aplicación en el entorno Heroku/Digital Ocean/Github.

Consideraciones Previas

- Se debe tener instalado el cliente heroku-cli: <https://devcenter.heroku.com/articles/heroku-cli#download-and-install>
- Se debe tener configurada la cuenta en Heroku: <https://dashboard.heroku.com/apps/smartdata-demo/deploy/heroku-git>
- Se debe contar con NodeJS y el gestor YARN instalados: <https://nodejs.org/es/download/> y <https://yarnpkg.com/en/docs/install>
- La aplicación a desplegar debe estar con GIT instalado y configurado

Paso a paso por terminal (consola)

Los siguientes son los pasos para realizar el deploy de una aplicación:

- Ingresar a la carpeta de la aplicación:
`cd ~/dev/cris/smartdata`
- Listar contenido de carpeta para verificar GIT instalado y configurado:
`ls -la`

En este caso, al existir la carpeta “.git” es la indicación que está configurado.

- verificar si ya está asociada la aplicación a Heroku:
`git remote -v`
 - * En este caso, al no estar asociada, NO muestra información de Heroku

Asociar la aplicación a heroku (en terminal o consola):

- `heroku login`

Con esta instrucción, se abrirá una página en el navegador para iniciar sesión en la cuenta de Heroku donde se desea hacer el deploy. Una vez iniciada la sesión, se presenta la página de confirmación. Una vez confirmada la sesión, se confirma también en el terminal. En este caso usando la cuenta de prueba (cas@cloded.com)

Luego de confirmar la sesión se debe agregar la configuración de Heroku

- `heroku git:remote -a smartdata-demo`

Ya es momento de hacer el Deploy en Heroku (despliegue), utilizando yarn.

Para simplificar este proceso y establecer las configuraciones necesarias, en la subcarpeta src/backend, se ha creado el Script deploy:heroku, ejecutado mediante NodeJS.

- cd src/backend
- yarn deploy:heroku

Una vez terminado el proceso de deploy, ya se puede visualizar la aplicación

- heroku open

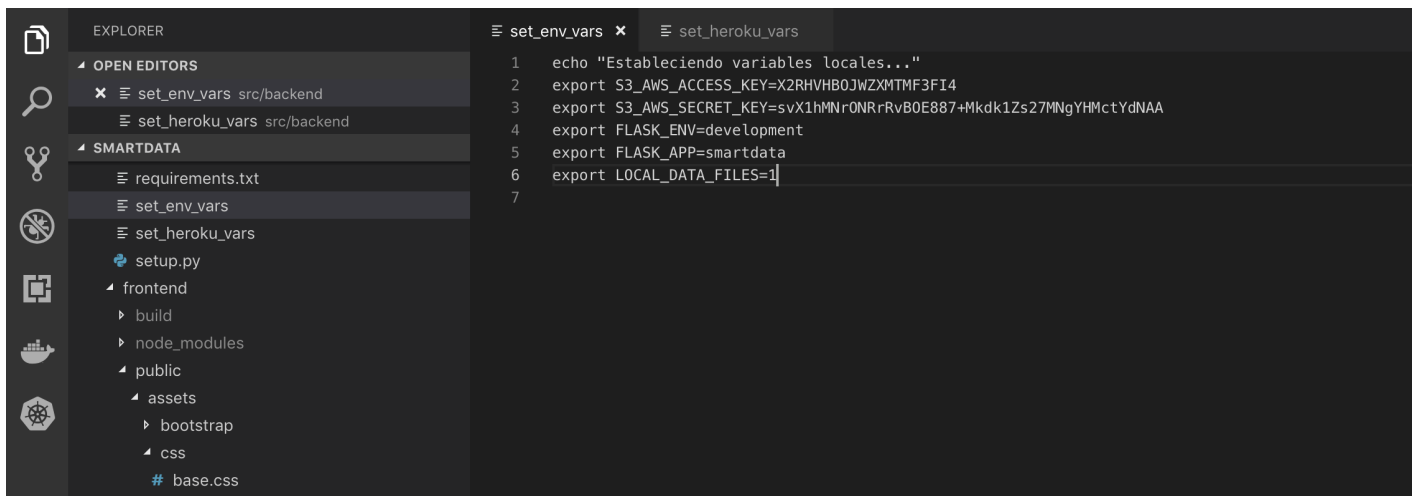
Conexión a archivos en Spaces

Desde backend Python/Flask

En la APP Flask, se configuran las credenciales en los archivos:

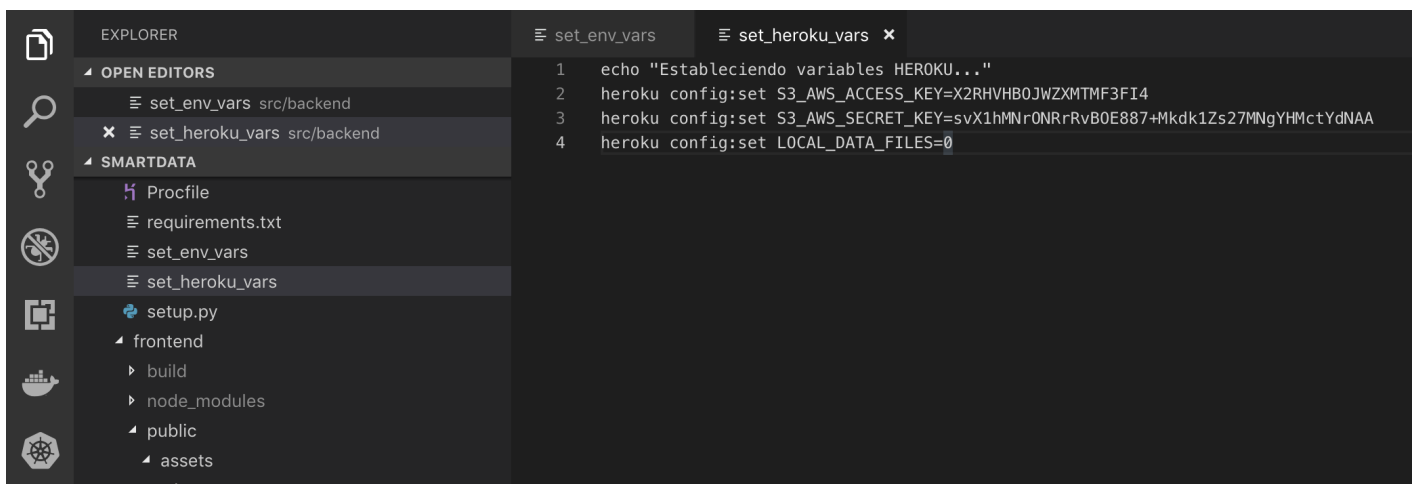
set_env_vars (para ambiente local)

set_heroku_vars (para ambiente Heroku)



The screenshot shows the VS Code interface. The Explorer sidebar on the left displays the project structure, including folders like 'frontend' and 'public', and files like 'requirements.txt', 'set_env_vars', and 'set_heroku_vars'. The Editor pane on the right shows the content of the 'set_env_vars' file, which contains shell commands to set environment variables for local development.

```
1 echo "Estableciendo variables locales..."
2 export S3_AWS_ACCESS_KEY=X2RHVHB0JWZXMTMF3FI4
3 export S3_AWS_SECRET_KEY=svX1hMNRrRvB0E887+MkdK1Zs27MNgYHMctYdNAA
4 export FLASK_ENV=development
5 export FLASK_APP=smartdata
6 export LOCAL_DATA_FILES=1
7
```



The screenshot shows the VS Code interface with the 'set_heroku_vars' file open in the Editor. The Explorer sidebar shows the same project structure. The Editor pane displays shell commands for configuring Heroku environment variables.

```
1 echo "Estableciendo variables HEROKU..."
2 heroku config:set S3_AWS_ACCESS_KEY=X2RHVHB0JWZXMTMF3FI4
3 heroku config:set S3_AWS_SECRET_KEY=svX1hMNRrRvB0E887+MkdK1Zs27MNgYHMctYdNAA
4 heroku config:set LOCAL_DATA_FILES=0
```

Las credenciales se establecen en las variables de entorno:

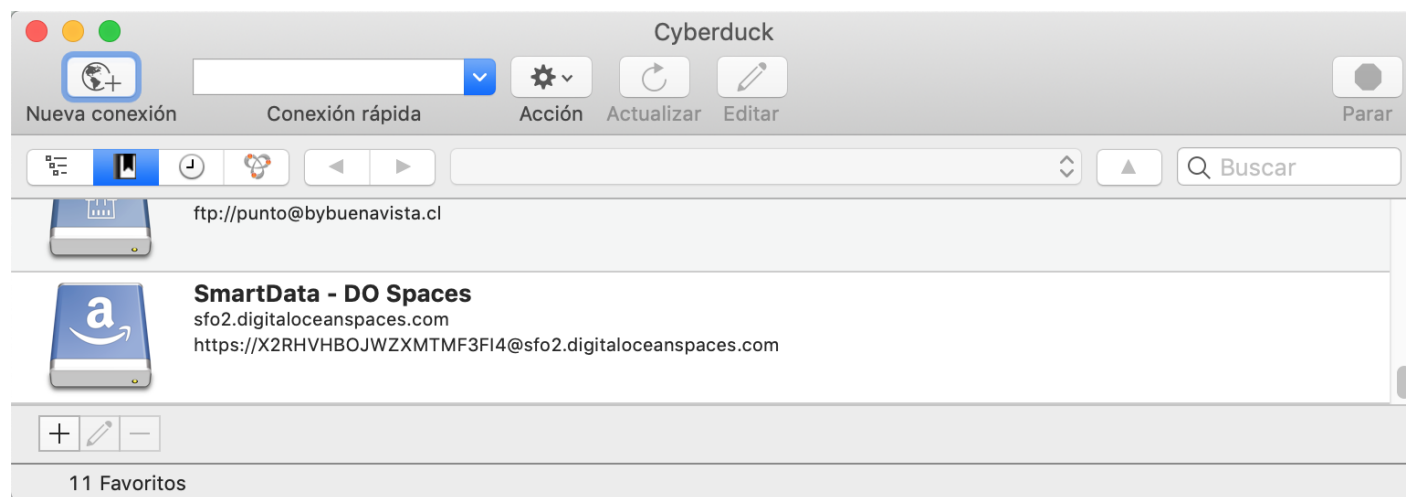
S3_AWS_ACCESS_KEY

S3_AWS_SECRET_KEY

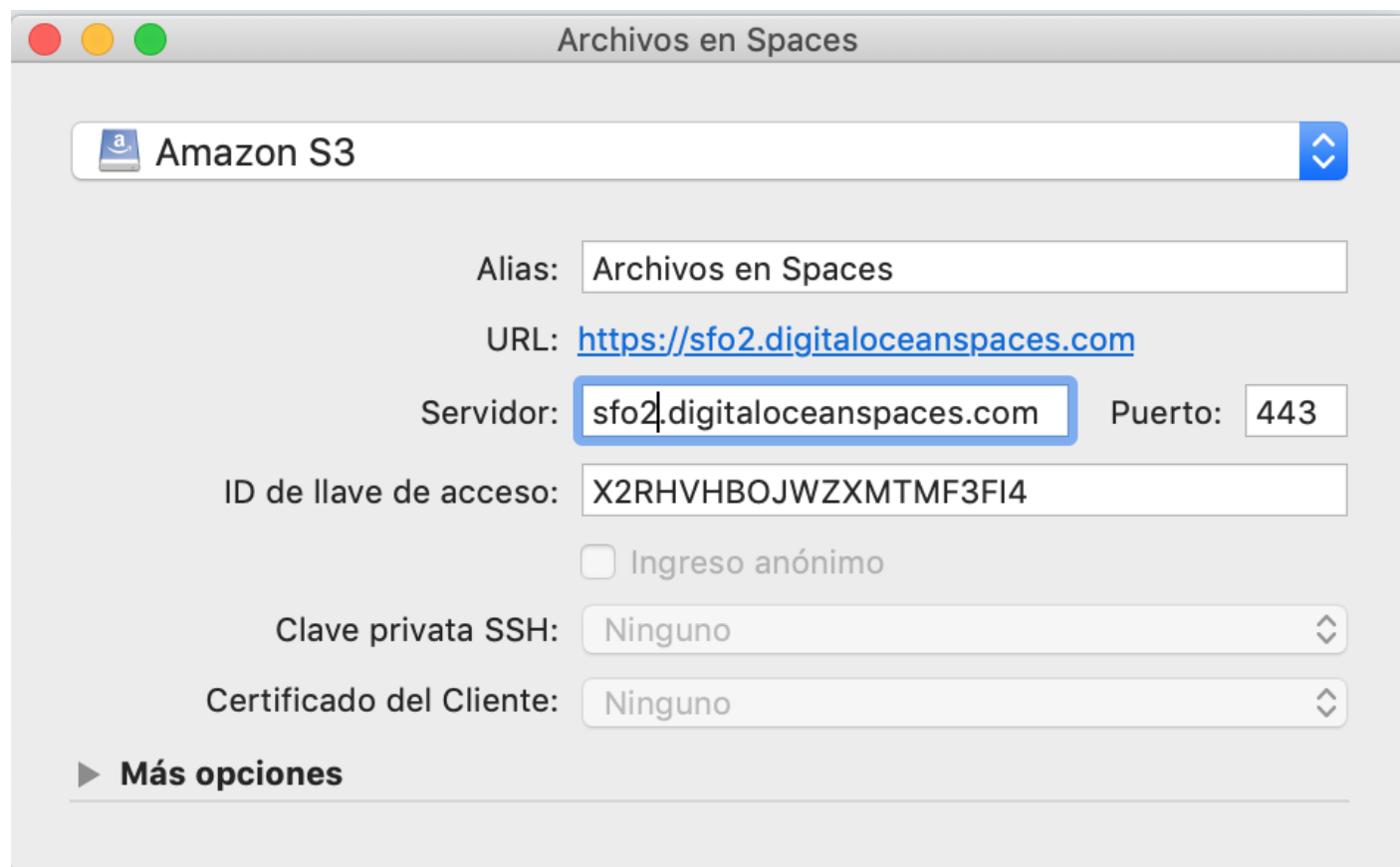
Desde cliente Cyberduck

Para conectar desde cliente, en este caso Cyberduck, se crea una nueva conexión, la que se puede agregar a favoritos, para no necesitar ingresar la información cada vez que se desea conectar.

Esto se realiza mediante el ícono “+” en la zona inferior de la ventana Cyberduck:



Abrirá la siguiente ventana



Se debe establecer la siguiente configuración

Alias: algún nombre que identifique la conexión

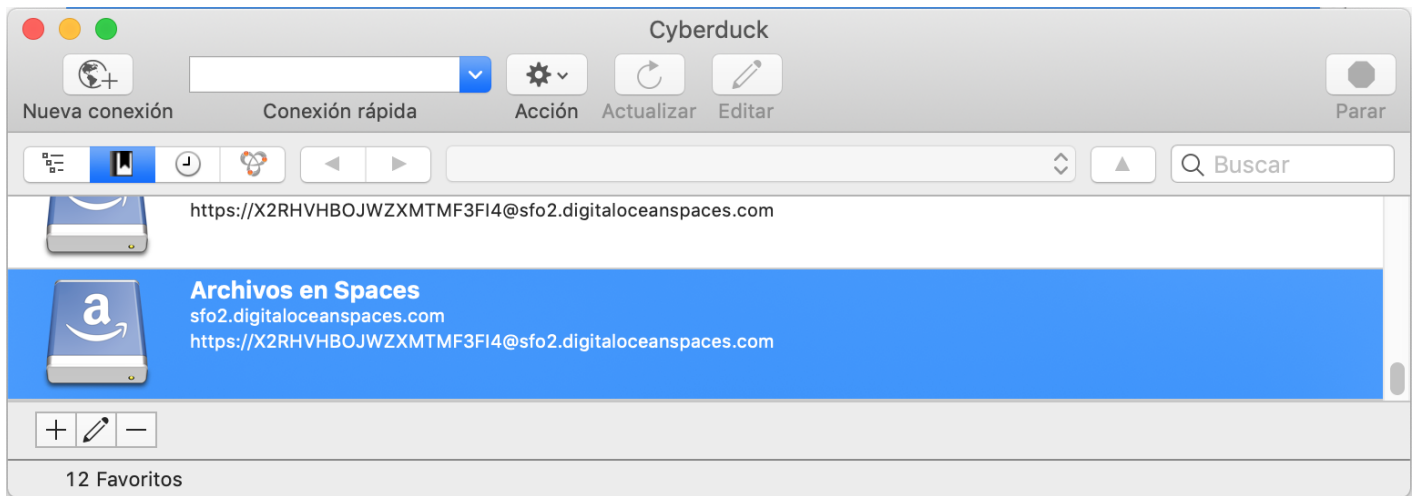
Servidor: sfo2.digitaloceanspaces.com

- Este valor dependerá de donde se establezca el Spaces en DigitalOcean

ID de llave de acceso: X2RHHVHBOJWZXMTMF3FI4

- Este valor lo entrega DigitalOcean al configurar una API para Spaces

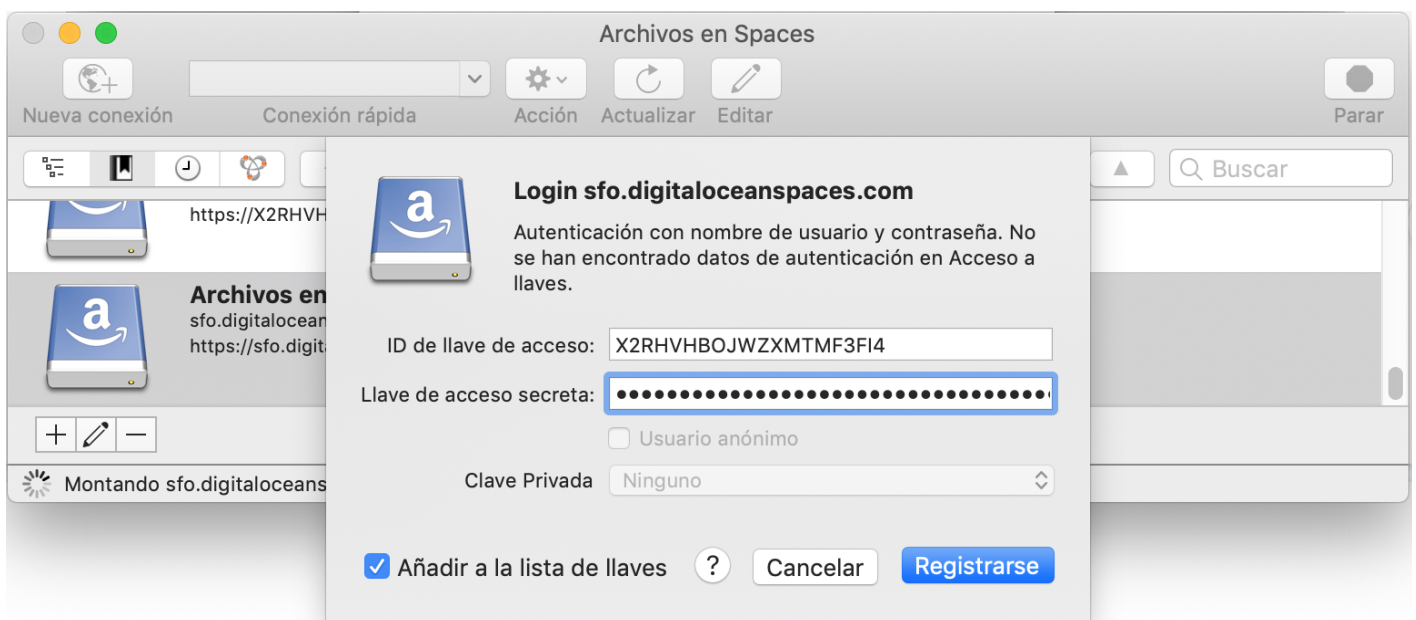
Luego se cierra la ventana y quedará registrada la conexión en favoritos



Se debe dar doble click al favorito e ingresar las credenciales de acceso para iniciar la conexión

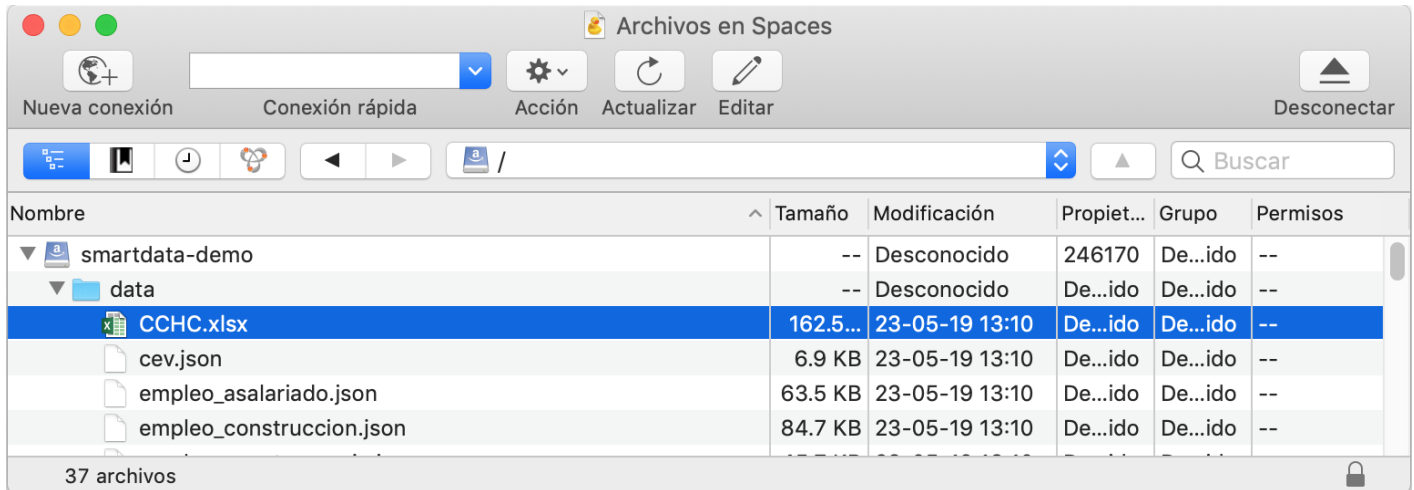
ID de llave de acceso: X2RHHVHBOJWZXMTMF3FI4

Llave de acceso secreta: svX1hMNRrONRrRvBOE887+Mkdk1Zs27MNgYHMctYdNAA



Click en “Registrarse” e iniciará la conexión.

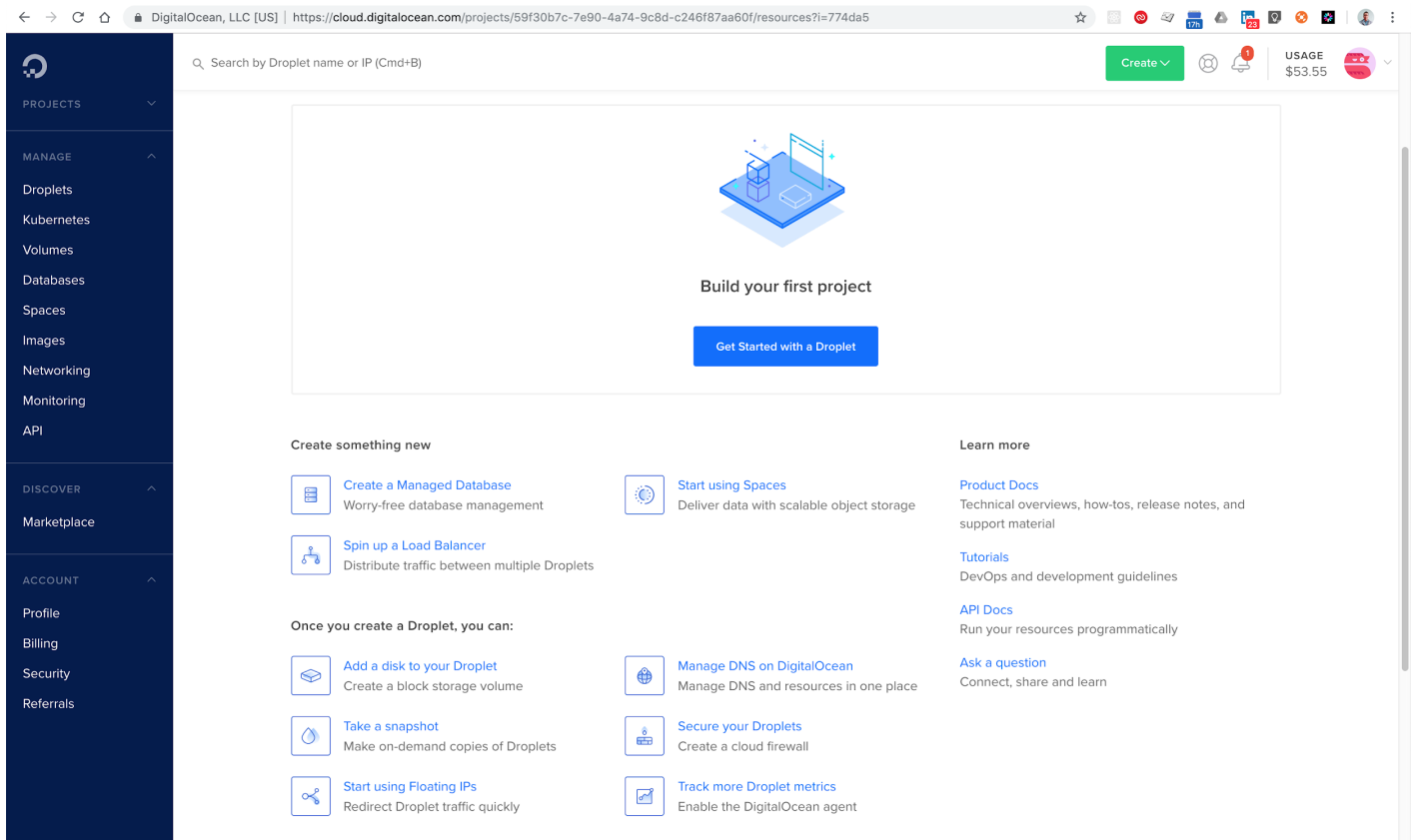
Se presentará la información del Spaces creado y la carpeta con archivos ubicados en “data”



En los siguientes puntos se indica como crear un Spaces en DigitalOcean y las credenciales necesarias.

Creación de Spaces y Credenciales en DigitalOcean

Una vez creada la cuenta en DigitalOcean, se debe crear un nuevo recurso Spaces en DigitalOcean



Se debe seleccionar una Zona donde ubicar el Spaces, por defecto se asigna la zona San Francisco 2, se recomienda usar alguna de USA.

Search by Droplet name or IP (Cmd+B)

Create

USAGE \$53.55

Create a Space

Choose a datacenter region

Region	Count
San Francisco	2
New York	3
Amsterdam	3
Singapore	1
Frankfurt	1

CDN (Content Delivery Network)

Deliver web assets up to 70% faster with global edge caching technology. Use a custom subdomain you own as an additional endpoint.

Enable CDN

Allow file listing?

☐ Restrict File Listing
Only users who connect to this Space using access keys can list the contents.

☒ Enable File Listing
Anyone can list the contents of this Space.

Important: This setting has no effect on whether individual files are visible. It only determines if anonymous users can list the name, size, and other metadata for the files in this Space.

Finalize and Create

Choose a unique name

Luego se indica “**sub dominio**” y proyecto al cual asignar el Spaces.

En este caso el sub dominio será “**smartdata-files**”, este se debe configurar en la APP Flask, en el archivo **files.py**

Search by Droplet name or IP (Cmd+B)

Create

USAGE \$53.55

Finalize and Create

Choose a unique name

Choose a unique subdomain name for your Space
smartdata-files ✓

Your Space's origin URL: <https://smartdata-files.sfo2.digitaloceanspaces.com>

Select project

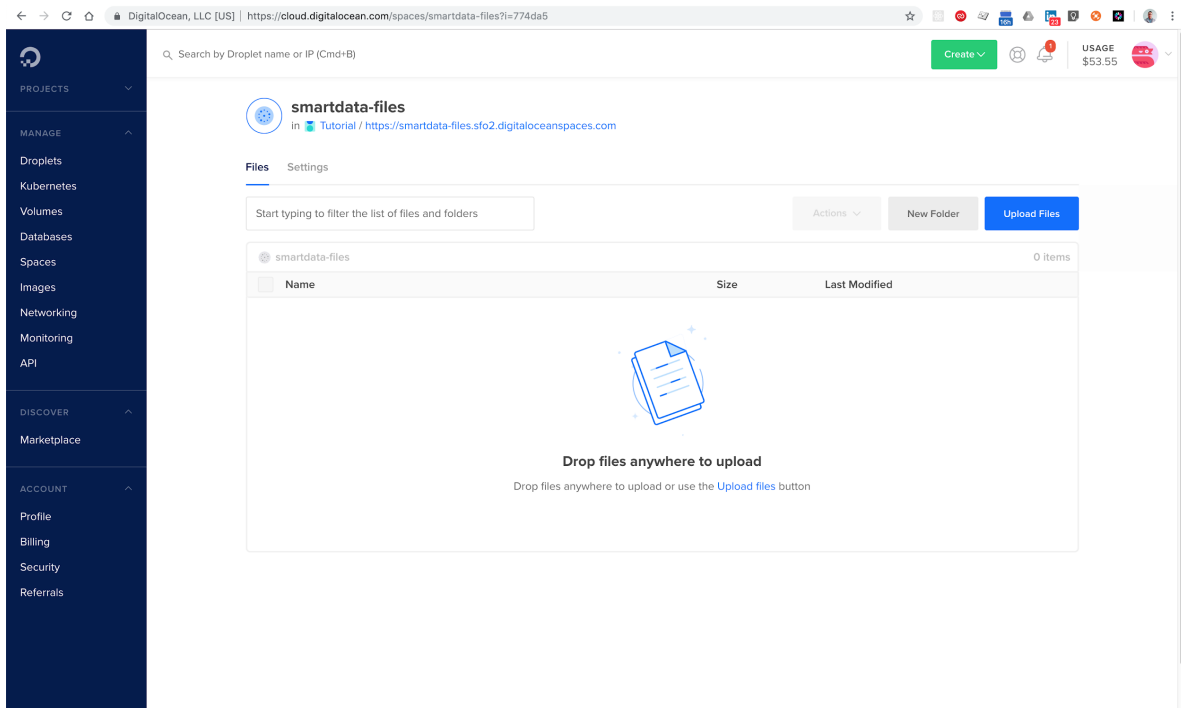
Select an existing project for this Space to belong to.

Tutorial

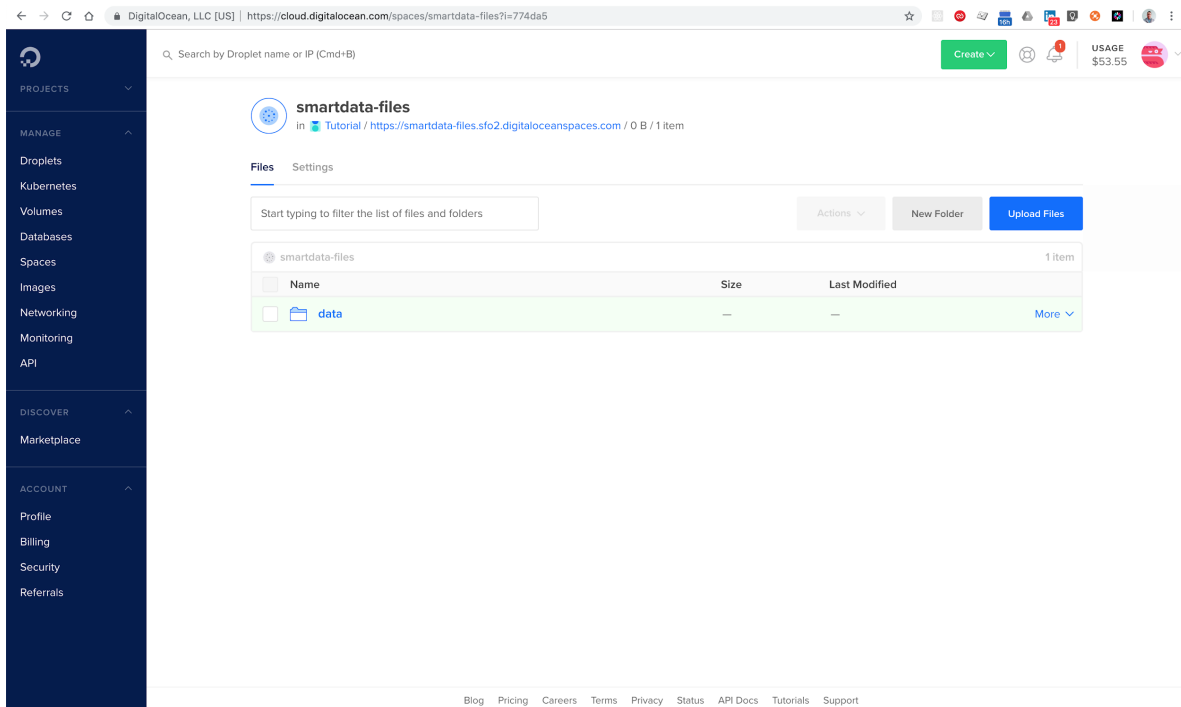
Create a Space

Blog Pricing Careers Terms Privacy Status API Docs Tutorials Support

Una vez creado el Spaces, se informa su creación.

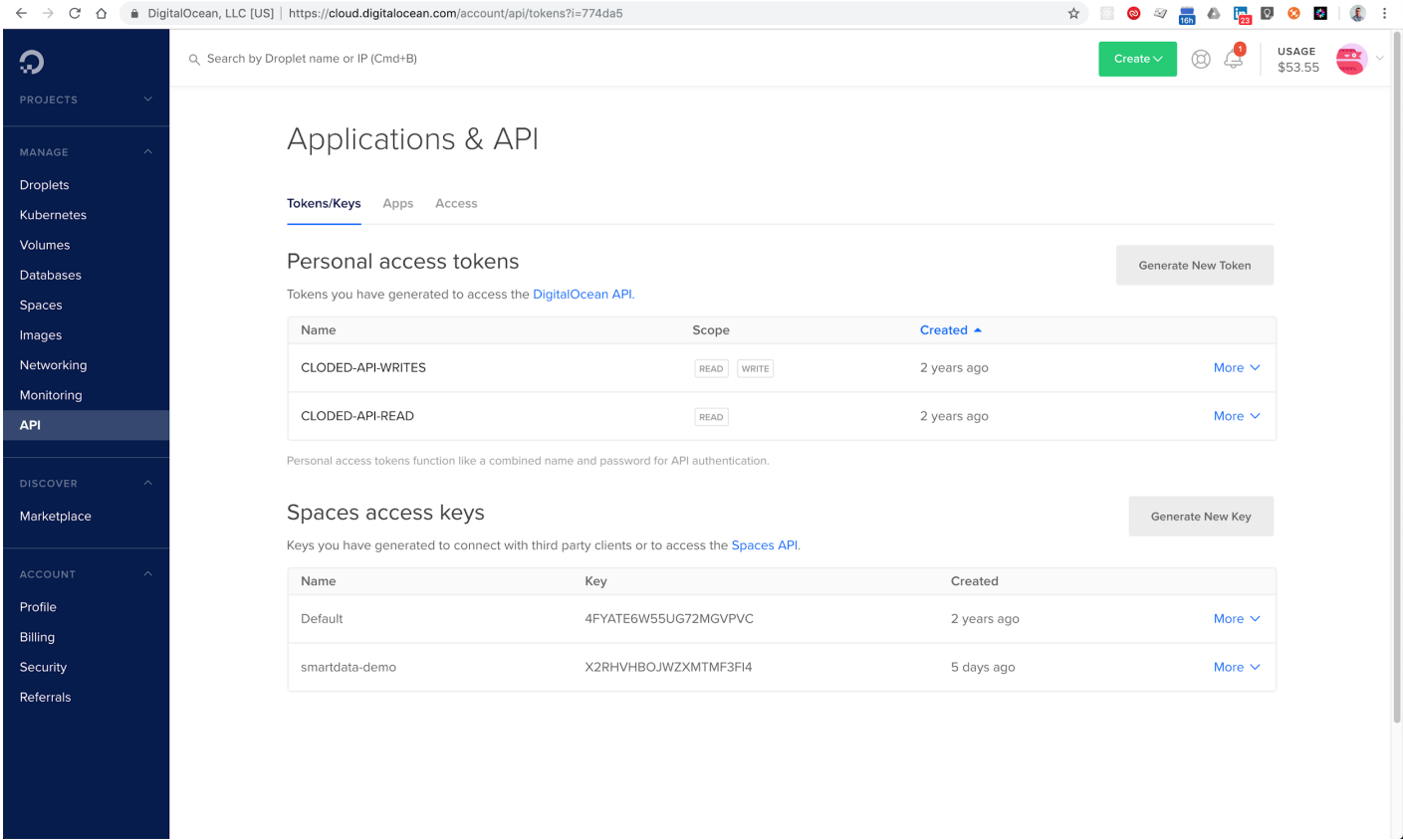


Se debe crear la carpeta “**data**” donde residirán los archivos.

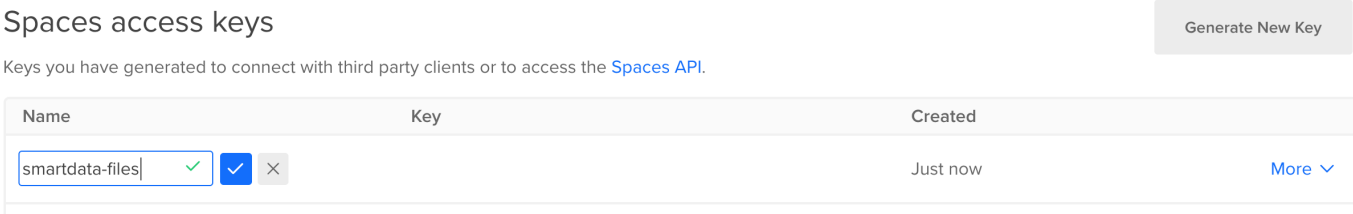


Creación de Credenciales de conexión

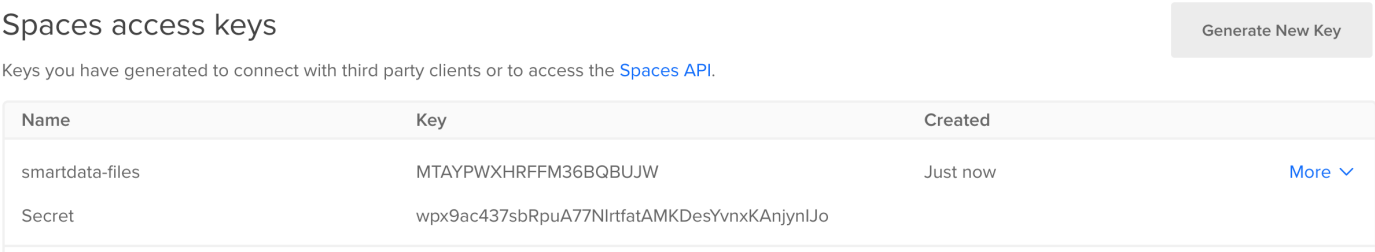
Para crear las credenciales necesarias, se debe acceder a “API” de DigitalOcean y ahí presionar “Generate New Key”



Luego se indica un nombre a la llave de conexión, en este caso “**smartdata-files**”

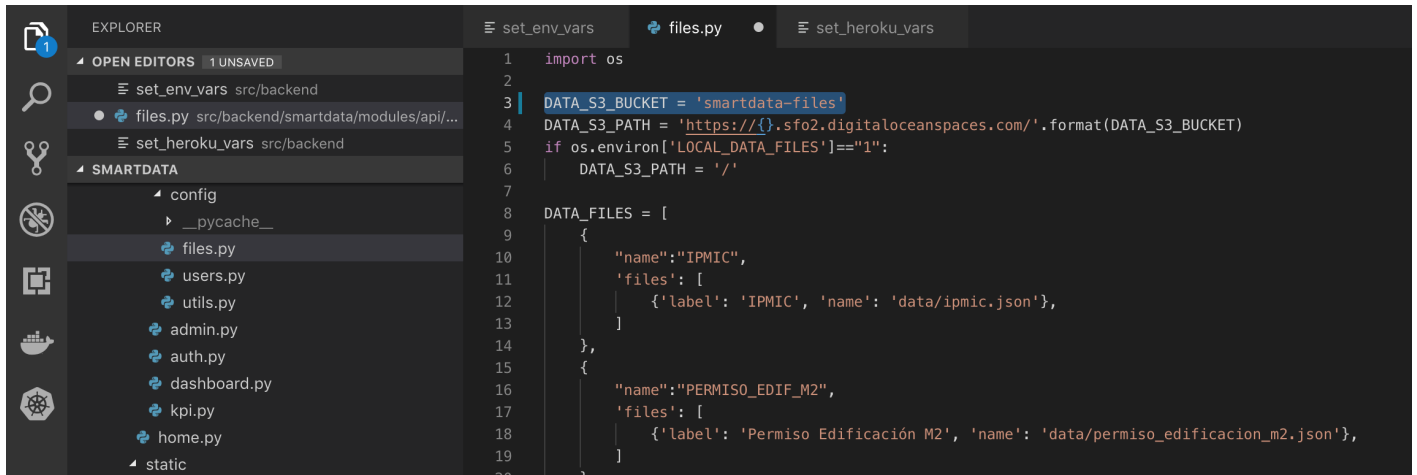


Al dar click al ticket, DigitalOcean creará las llaves de conexión



En caso de cambiar o crear un nuevo Spaces o credenciales, se debe “re configurar” la aplicación Python/Flask.

Se usa el sub dominio en la configuración de la aplicación Python/Flask en “files.py”

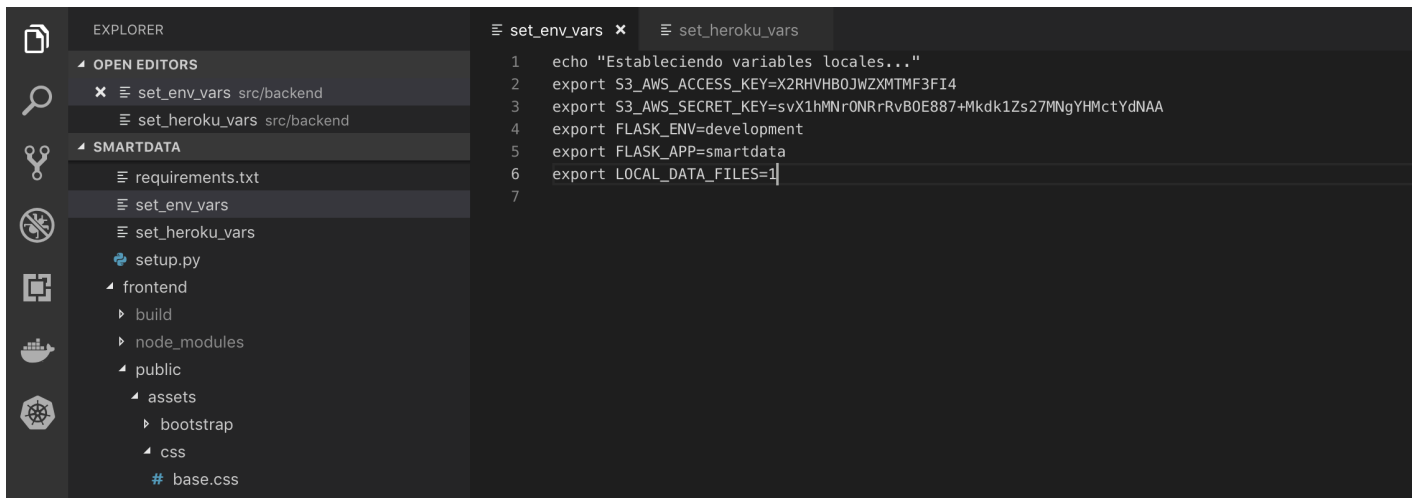


```
1 import os
2
3 DATA_S3_BUCKET = 'smartdata-files'
4 DATA_S3_PATH = 'https://{}.sfo2.digitaloceanspaces.com/'.format(DATA_S3_BUCKET)
5 if os.environ['LOCAL_DATA_FILES']=="1":
6     DATA_S3_PATH = '/'
7
8 DATA_FILES = [
9     {
10         "name": "IPMIC",
11         'files': [
12             {'label': 'IPMIC', 'name': 'data/ipmic.json'},
13         ]
14     },
15     {
16         "name": "PERMISO_EDIF_M2",
17         'files': [
18             {'label': 'Permiso Edificación M2', 'name': 'data/permiso_edificacion_m2.json'},
19         ]
20     }
21 ]
```

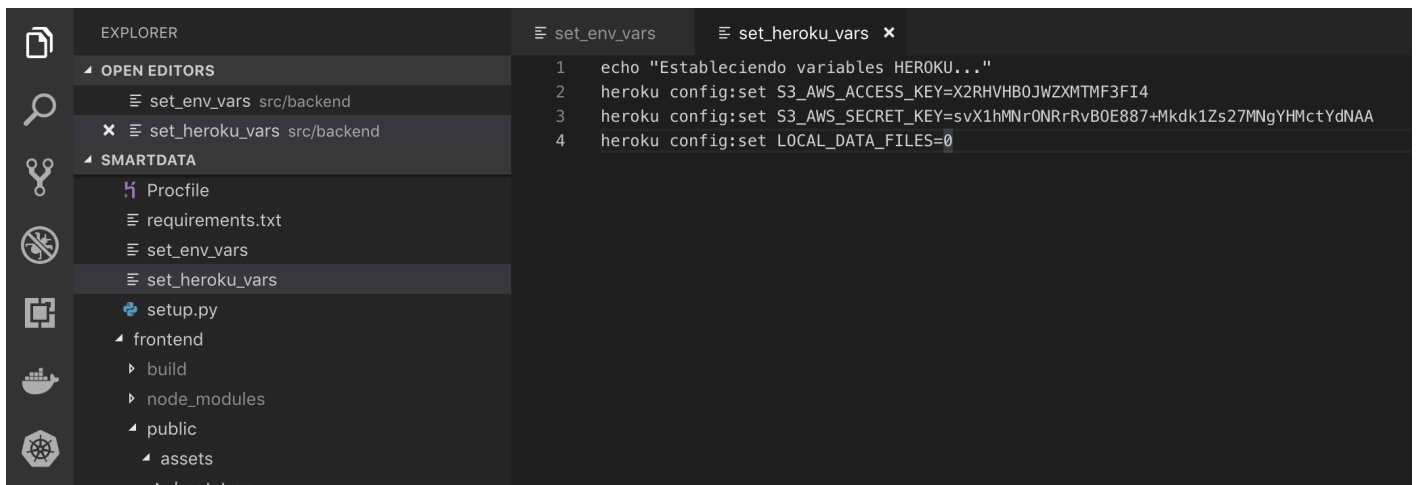
Se configuran las credenciales en los archivos:

set_env_vars (para ambiente local)

set_heroku_vars (para ambiente Heroku)



```
1 echo "Estableciendo variables locales..."
2 export S3_AWS_ACCESS_KEY=X2RHVHB0JWZXMTMF3FI4
3 export S3_AWS_SECRET_KEY=svX1hMNrONRrRvB0E887+MkdK1Zs27MNgYHMctYdNAA
4 export FLASK_ENV=development
5 export FLASK_APP=smartdata
6 export LOCAL_DATA_FILES=1
7
```



```
1 echo "Estableciendo variables HEROKU..."
2 heroku config:set S3_AWS_ACCESS_KEY=X2RHVHB0JWZXMTMF3FI4
3 heroku config:set S3_AWS_SECRET_KEY=svX1hMNrONRrRvB0E887+MkdK1Zs27MNgYHMctYdNAA
4 heroku config:set LOCAL_DATA_FILES=0
```