

DAWBIO2- M08 UF3

Pt. Hospedaje de una aplicación web con Azure App Service

DAWBIO2- M08 UF3	1
Pt. Hospedaje de una aplicación web con Azure App Service	1
1. Una vez hemos creado nuestra cuenta y hemos creado una instancia gratuita, vamos a bajar el servicio Azure a nuestro local.	2
2. Ahora que ya tenemos la sesión viva, vamos a crear una aplicación pequeña con Python y Flask.	4
3. Creamos el fichero app.py con el código de la app. dentro. (Ve el contenido en el imagen.)	5
4. Extraemos las dependencias al "requirements.txt".	5
5. Vamos a ejecutar la app para comprobar.	5
6. Ahora ya podemos intentar desplegar nuestra app al Azura.	6
7. Una vez el deployment ha acabado, comprobamos nuestro dominio si la app ha sido desplegado con éxito.	7

1. Una vez hemos creado nuestra cuenta y hemos creado una instancia gratuita, vamos a bajar el servicio Azure a nuestro local.

Ejecutamos los siguientes comandos:

- `curl -L https://aka.ms/InstallAzureCli | bash`

```
(venv) ~/workspace:
$ curl -L https://aka.ms/InstallAzureCli | bash
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left     Speed
100 1405 100 1405    0     0 1207      0  0:00:01  0:00:01 --:--:-- 3881
Downloading Azure CLI install script from https://azurecliprod.blob.core.windows.net/install.py to /tmp/azure_cli_in
all_tmp_02ejDs.
##### 100.0%
/tmp/azure_cli_install_tmp_02ejDs: OK
Running install script.
-- Verifying Python version.
-- Python version 3.10.6 okay.
-- Verifying native dependencies.
-- Unable to verify native dependencies. dist=pop!_os, version=22.04. Continuing...

==> In what directory would you like to place the install? (leave blank to use '/home/dama/lib/azure-cli'): 
```

Este commando nos va a instalar todo del entorno que necesitamos para desplegar la aplicación y nos va a instalar las dependencias de Python también con PIP.

```
Collecting urllib3[secure_extra]
  Downloading urllib3_secure_extra-0.1.0-py2.py3-none-any.whl (1.4 kB)
Collecting pycparser
  Downloading pycparser-2.21-py2.py3-none-any.whl (118 kB)
    |#####| 118.7/118.7 kB 5.9 MB/s eta 0:00:00
Collecting pygments
  Downloading Pygments-2.14.0-py3-none-any.whl (1.1 MB)
    |#####| 1.1/1.1 MB 5.9 MB/s eta 0:00:00
Collecting pyyaml
  Downloading PyYAML-6.0-cp310-cp310-manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_12_x86_64.manylinux2010_x86_
.whl (682 kB)
    |#####| 682.2/682.2 kB 6.5 MB/s eta 0:00:00
Collecting tabulate
  Downloading tabulate-0.9.0-py3-none-any.whl (35 kB)
Collecting requests-oauthlib>=0.5.0
  Downloading requests_oauthlib-1.3.1-py2.py3-none-any.whl (23 kB)
Collecting bcrypt>=3.1.3
  Downloading bcrypt-4.0.1-cp36-abi3-manylinux_2_28_x86_64.whl (593 kB)
    |#####| 593.7/593.7 kB 6.6 MB/s eta 0:00:00
Collecting charset-normalizer<4,>=2
  Downloading charset_normalizer-3.0.1-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (198 kB)
    |#####| 198.8/198.8 kB 4.4 MB/s eta 0:00:00
```

- `source .bashrc` (Actualizamos los cambios hecho en nuestro fichero de configuración de Bash.

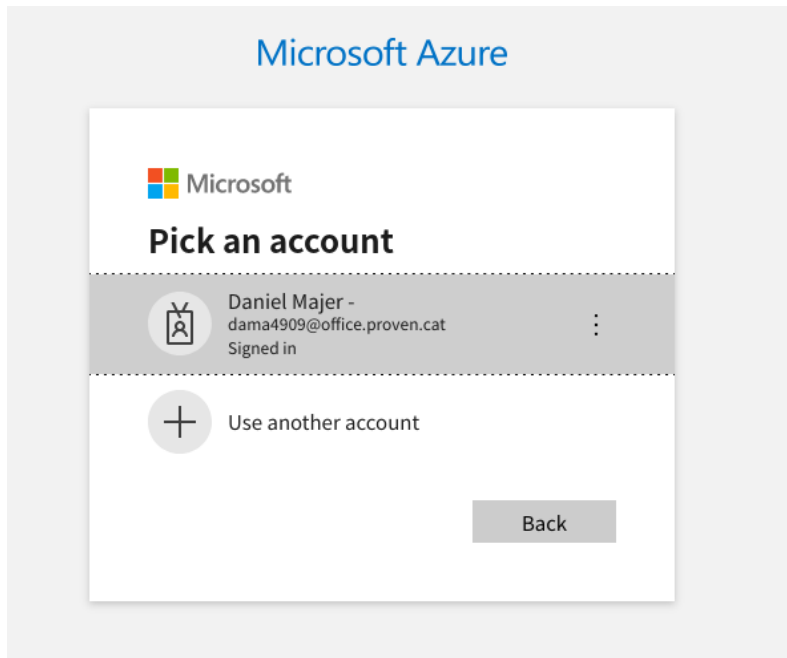
```
(venv) dama@pop-os:~$
$
(venv) dama@pop-os:~$
$ source .bashrc
(venv) dama@pop-os:~$
$
```

Una vez lo tenemos hecho la instalación en nuestra máquina local, vamos a intentar realizar un login:

-az login

```
(venv) dama@pop-os:~$ az login
$ az login
A web browser has been opened at https://login.microsoftonline.com/organizations/oauth2/v2.0/authorize. Please continue the login in the web browser. If no web browser is available or if the web browser fails to open, use device code flow with 'az login --use-device-code'.
```

Que nos abra el navegador para login.



Hacemos click a nuestra cuenta.

You have logged into Microsoft Azure!

You can close this window, or we will redirect you to the [Azure CLI documentation](#) in 10 seconds.

En el terminal nos va a salir un JSON de detalles sobre nuestra sesión.

```

ow with az login --use-device-code .
[
  {
    "cloudName": "AzureCloud",
    "homeTenantId": "836f1d43-90b9-41eb-815f-6e37bd65ff30",
    "id": "9d168eac-4f12-4019-aabf-4c08a9ac9766",
    "isDefault": true,
    "managedByTenants": [],
    "name": "Azure for Students",
    "state": "Enabled",
    "tenantId": "836f1d43-90b9-41eb-815f-6e37bd65ff30",
    "user": {
      "name": "dama4909@office.proven.cat",
      "type": "user"
    }
  }
]

```

2. Ahora que ya tenemos la sesión viva, vamos a crear una aplicación pequeña con Python y Flask.

Creemos un entorno virtual nuevo, lo activamos y instalamos Flask.

- deactivate (Solo si ya tenemos un entorno activado.)
- python3 -m venv venv
- source ./venv/bin/activate
- pip install Flask

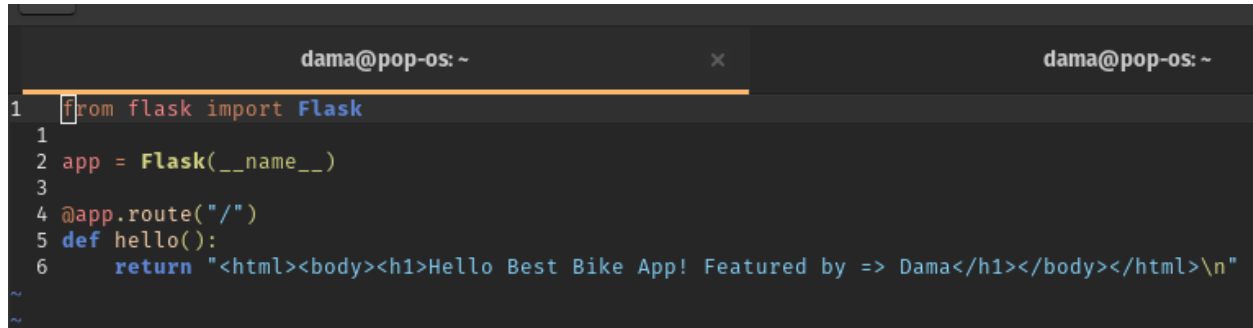
```

dama@pop-os:~
$ python -m venv venv
Command 'python' not found, did you mean:
  command 'python3' from deb python3
  command 'python' from deb python-is-python3
dama@pop-os:~
$ python3 -m venv venv
dama@pop-os:~
$ source venv/bin/activate
(venv) dama@pop-os:~
$ pip install Flask
Collecting Flask
  Using cached Flask-2.2.2-py3-none-any.whl (101 kB)
Collecting itsdangerous>=2.0
  Using cached itsdangerous-2.1.2-py3-none-any.whl (15 kB)
Collecting Werkzeug>=2.2.2
  Using cached Werkzeug-2.2.2-py3-none-any.whl (232 kB)
Collecting Jinja2>=3.0
  Using cached Jinja2-3.1.2-py3-none-any.whl (133 kB)

```

3. Creamos el fichero app.py con el código de la app. dentro.
(Ve el contenido en el imagen.)

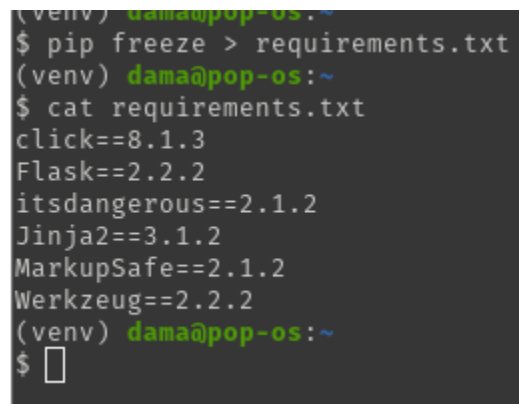
- nvim app.py



```
dama@pop-os: ~  
1 from flask import Flask  
2 app = Flask(__name__)  
3  
4 @app.route("/")  
5 def hello():  
6     return "<html><body><h1>Hello Best Bike App! Featured by => Dama</h1></body></html>\n"
```

4. Extraemos las dependencias al "requirements.txt".

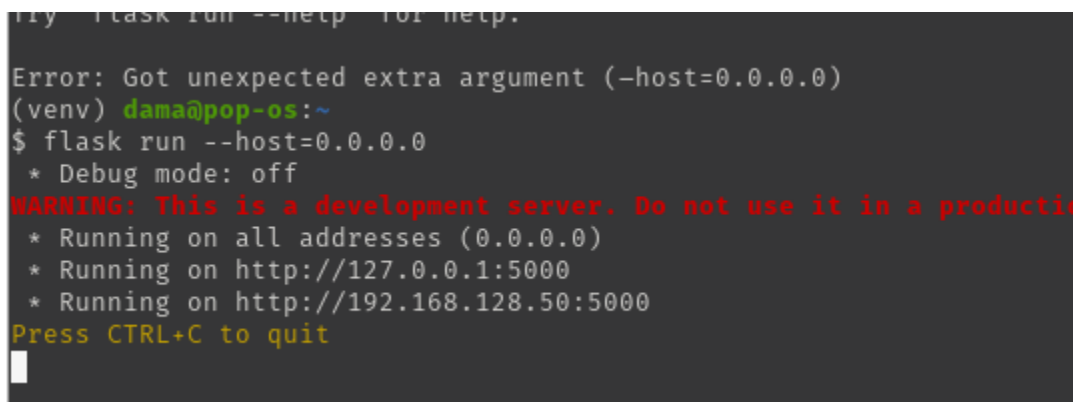
- pip freeze > requirements.txt



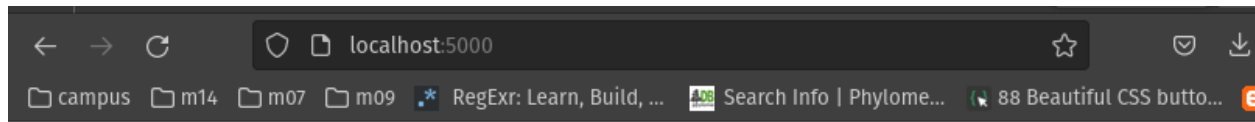
```
(venv) dama@pop-os:~  
$ pip freeze > requirements.txt  
(venv) dama@pop-os:~  
$ cat requirements.txt  
click==8.1.3  
Flask==2.2.2  
itsdangerous==2.1.2  
Jinja2==3.1.2  
MarkupSafe==2.1.2  
Werkzeug==2.2.2  
(venv) dama@pop-os:~  
$
```

5. Vamos a ejecutar la app para comprobar.

- flask run --host=0.0.0.0



```
Try "flask run --help" for help.  
Error: Got unexpected extra argument (-host=0.0.0.0)  
(venv) dama@pop-os:~  
$ flask run --host=0.0.0.0  
* Debug mode: off  
WARNING: This is a development server. Do not use it in a production  
* Running on all addresses (0.0.0.0)  
* Running on http://127.0.0.1:5000  
* Running on http://192.168.128.50:5000  
Press CTRL+C to quit
```



Hello Best Bike App! Featured by => Dama

La app en local funciona!

6. Ahora ya podemos intentar desplegar nuestra app al Azura.

Tenemos que comprobar la lista de Azure:

- az webapp list -o table

```
(venv) dama@pop-os:~$ az webapp list -o table
Name                                Location    State    ResourceGroup    DefaultHostName                                AppServicePlan
-----
dama-dawbio2                        West Europe Running    dama-dawbio2_group    dama-dawbio2.azurewebsites.net    ASP-damadawbio2group-87bf
(venv) dama@pop-os:~$
```

Como podemos ver nuestra instancia de Azure aparece.

(Arreglación!) Creamos un directorio para nuestro fichero de app.py y requirements.txt.

- mkdir ~/azure_test
- mv -t azure_test app.py requirements.txt
- cd azure_test/

```
-rw----- 1 dama dama      52 Dec 19 19:02 .Xauthority
(venv) dama@pop-os:~$ mkdir ~/azure_test
(venv) dama@pop-os:~$ mv -t azure_test app.py requirements.txt
(venv) dama@pop-os:~$ cd azure_test/
(venv) dama@pop-os:~/azure_test$
(venv) dama@pop-os:~/azure_test$
```

Vamos a exportar unos variables globales que Azure necesita en nuestro local.

- export APPNAME=\$(az webapp list --query [0].name --output tsv)
- export APPRG=\$(az webapp list --query [0].resourceGroup --output tsv)
- export APPPLAN=\$(az appservice plan list --query [0].name --output tsv)
- export APPSKU=\$(az appservice plan list --query [0].sku.name --output tsv)
- export APPLOCATION=\$(az appservice plan list --query [0].location --output tsv)

Arrancamos el proceso de despliegament.

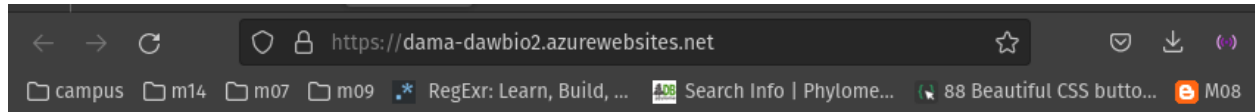
- az webapp up --name \$APPNAME --resource-group \$APPRG --plan \$APPPLAN --sku \$APPSKU --location "\$APPLOCATION"

Ahora esperamos para Azure que se termine el deployment.

```
(venv) dama@pop-os:~/azure_test
$ export APPNAME=$(az webapp list --query [0].name --output tsv)
az: command not found
(venv) dama@pop-os:~/azure_test
$ source ~/.bashrc
(venv) dama@pop-os:~/azure_test
$ export APPNAME=$(az webapp list --query [0].name --output tsv)
(venv) dama@pop-os:~/azure_test
$ export APPRG=$(az webapp list --query [0].resourceGroup --output tsv)
(venv) dama@pop-os:~/azure_test
$ export APPPLAN=$(az appservice plan list --query [0].name --output tsv)
(venv) dama@pop-os:~/azure_test
$ export APPSKU=$(az appservice plan list --query [0].sku.name --output tsv)
(venv) dama@pop-os:~/azure_test
$ export APPLOCATION=$(az appservice plan list --query [0].location --output tsv)
(venv) dama@pop-os:~/azure_test
$ az webapp up --name $APPNAME --resource-group $APPRG --plan $APPPLAN --sku $APPSKU --location $APPLOCATION
Webapp 'dama-dawbio2' already exists. The command will deploy contents to the existing app.
Creating AppServicePlan 'ASP-damadawbio2group-87bf' ...
Creating zip with contents of dir /home/dama/azure_test ...
Getting scm site credentials for zip deployment
Starting zip deployment. This operation can take a while to complete ...
```

7. Una vez el deployment ha acabado, comprobamos nuestro dominio sí la app ha sido desplegado con éxito.

```
Starting zip deployment. This operation can take a while to complete ...
Deployment endpoint responded with status code 202
You can launch the app at http://dama-dawbio2.azurewebsites.net
Setting 'az webapp up' default arguments for current directory. Manage defaults with 'az config webapp up'
--resource-group/-g default: dama-dawbio2_group
--sku default: F1
--plan/-p default: ASP-damadawbio2group-87bf
--location/-l default: westeurope
--name/-n default: dama-dawbio2
{
  "URL": "http://dama-dawbio2.azurewebsites.net",
  "appserviceplan": "ASP-damadawbio2group-87bf",
  "location": "westeurope",
  "name": "dama-dawbio2",
  "os": "Linux",
  "resourcegroup": "dama-dawbio2_group",
  "runtime_version": "python|3.10",
  "runtime_version_detected": "-",
  "sku": "FREE",
  "src_path": "//home//dama//azure_test"
}
(venv) dama@pop-os:~/azure_test
$
```



Hello Best Bike App! Featured by => Dama

Link de la app: <https://dama-dawbio2.azurewebsites.net/>

El deployment ha sido realizado con exito! Ya podemos solicitar trabajos devops! 😊