



CEU

| *Universidad
San Pablo*

Despliegue en la nube con Azure
Aplicación: UniConnect

(MacOS o Linux)



Miembros: Dulibeth Medina, Marta Canino, Yolanda Llop, Xavier Mora y Juan Grima.

Índice

Introducción	3
Configuración del entorno virtual	3
1. Clonar el repositorio	3
2. Crear entorno virtual	3
2. Activar entorno virtual	3
3. Instalar dependencias	3
4. Ejecución de aplicación	4
Despliegue con Azure	4
1. Crear una cuenta en Azure	4
2. Descargar CLI de azure	4
3. Inicio de sesión en Azure	4
4. Ejecución del despliegue	5

Introducción

Esta guía brinda pasos detallados sobre cómo hemos implementado la aplicación web hecha con Python y Flask UniConnect en la nube de Azure.

Al seguir esta documentación, ganarás una comprensión clara de los requisitos previos, configuraciones necesarias y los pasos específicos que hemos llevado a cabo para lograr un despliegue exitoso de UniConnect en Azure.

Esta guía es válida para realizar el despliegue tanto en MacOS como en Linux.

Configuración del entorno virtual

Para implementar la aplicación escrita en Python con el framework Flask, es fundamental configurar un entorno virtual como primer paso antes de llevar a cabo el despliegue.

1. Clonar el repositorio

En este paso hemos clonado el repositorio de la aplicación en un directorio de despliegue.

```
deploy-azure % git clone https://github.com/XMoraP/Practica-Ing.software.git
```

2. Crear entorno virtual

Hemos creado un entorno virtual dentro del proyecto con el siguiente comando:

```
Practica-Ing.software % python3 -m venv .venv
```

2. Activar entorno virtual

Luego de crear el entorno virtual, lo hemos activado con el siguiente comando:

```
Practica-Ing.software % source .venv/bin/activate
```

3. Instalar dependencias

Dentro del entorno hemos instalado las dependencias necesarias en el fichero `'requirements.txt'` con el comando:

```
Practica-Ing.software % pip install -r requirements.txt
```

4. Ejecución de aplicación

Con el entorno virtual ya configurado procedemos a probar aplicación desde el entorno.

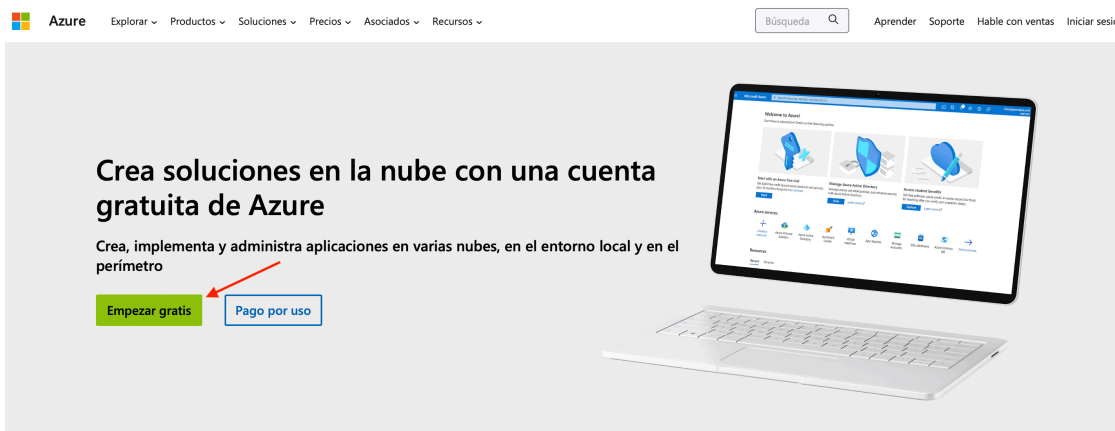
```
Practica-Ing.software % flask run
```

Si la aplicación funciona bien procedemos a realizar el despliegue.

Despliegue con Azure

1. Crear una cuenta en Azure

Entramos en el enlace [azure](#) y creamos una cuenta en Azure de forma gratuita.



2. Descargar CLI de azure

Para realizar el despliegue con Azure es necesario disponer de la línea de comandos en nuestra máquina (Azure CLI). Se puede descargar con los siguientes comandos:

- **MacOS:**

```
Practica-Ing.software % brew update && brew install azure-cli
```

- **Linux:**

```
curl -sL https://aka.ms/InstallAzureCLIDeb | sudo bash
```

3. Inicio de sesión en Azure

Una vez creada la cuenta y disponer de la CLI, iniciamos sesión desde la consola ejecutando el comando:

```
Practica-Ing.software % az login
```

Al ejecutar este comando, se abrirá el navegador para iniciar sesión en Azure. Si la sesión se inicia correctamente, verás lo siguiente en la consola:

```
[
  {
    "cloudName": "AzureCloud",
    "homeTenantId": "34330eef-31be-4cd1-925e-a790ac722d30",
    "id": "a9391420-e1b0-4950-b5ed-8d607bd5184f",
    "isDefault": true,
    "managedByTenants": [],
    "name": "Azure for Students",
    "state": "Enabled",
    "tenantId": "34330eef-31be-4cd1-925e-a790ac722d30",
    "user": {
      "name": "xavieralexander.moraperaza@usp.ceu.es",
      "type": "user"
    }
  }
]
```

4. Ejecución del despliegue

El despliegue lo hemos realizado con la ejecución del comando **'az webapp up'**. El comando dispone de una serie de parámetros configurables.

➔ Parámetros del comando:

- **Az webapp up:** Para realizar un despliegue de una aplicación web.
- **--runtime:** Para indicar que versión de Python se está utilizando.
- **--sku:** Para indicar el plan que se quiere contratar. En nuestro caso **'F1'**, el cual indicará que se quiere contratar el plan **gratuito**.
- **--logs:** Para habilitar la visualización de la secuencia de registros.
- **--name:** Para indicar el nombre de la aplicación, el cual debe ser único.

➔ Comando utilizado:

```
Practica-Ing.software % az webapp up --runtime PYTHON:3.9 --sku F1 --logs --name uni-connectt
```

Si todo va bien aparece por consola:

```
The webapp 'uni-connectt' doesn't exist
Creating Resource group 'xavieralexander.moraperaza_rg_8495' ...
Resource group creation complete
Creating AppServicePlan 'xavieralexander.moraperaza_asp_6631' or Updating if already exists
Creating webapp 'uni-connectt' ...
Configuring default logging for the app, if not already enabled
Creating zip with contents of dir /Users/xaviermoraperaza/workSpace/deploy-azure/Practica-Ing.software ...
Getting scm site credentials for zip deployment
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://uni-connectt.azurewebsites.net
Configuring default logging for the app, if not already enabled
```

Si el despliegue ha sido **exitoso** sale por consola:

```
2023-11-13T21:40:48 Welcome, you are now connected to log-streaming service.
```