



Objetivo

Aprender a conectar un proyecto local desarrollado con Django (backend) y React (frontend) en Visual Studio Code con los servicios de AWS, específicamente utilizando Elastic Beanstalk para desplegar el backend y S3 o Amplify para desplegar el frontend.

Práctica 2.2: Conexión entre proyecto local y servicios de AWS (VS Code con Django y React)

● ¿Qué es lo que haremos?

- 1 Crear un proyecto simple con Django y React.
 - 2 Subir el backend (Django) a **Elastic Beanstalk**, un servicio de AWS que ejecuta aplicaciones web automáticamente.
 - 3 Subir el frontend (React) a **AWS S3** o **AWS Amplify**, servicios que permiten mostrar páginas web estáticas.
- Esto permitirá que tu proyecto se vea desde cualquier parte del mundo a través de internet, sin necesidad de mantener un servidor físico propio.

● Requisitos previos:

- ➔ Tener Visual Studio Code instalado.
- ➔ Tener Python, Node.js y Git instalados.
- ➔ Tener una cuenta activa en AWS (ya hecha en la práctica 2.1).
- ➔ Haber configurado correctamente la CLI de AWS (se explicará aquí también).



Paso 1: Crear el proyecto local en VS Code.

- Generar los archivos iniciales de un proyecto con Django para el backend y React para el frontend, para tener una aplicación local con lógica del lado del servidor (Django) y del lado del cliente (React).
- Instrucciones:
- Crear carpeta base del proyecto
 - 1 Con tu compañero de la práctica 13, comiencen la configuración de su entorno como backend y Frontend, al igual que crear su repositorio en Github o utilicen el mismo proyecto de la práctica 1.3
 - 2 Abre Visual Studio Code y abre tu proyecto.
 - 3 Abre una terminal de VS Code (Terminal > Nueva terminal).
- Crear entorno virtual para Django

```
python -m venv venv
source venv/bin/activate # En Linux/macOS
venv\Scripts\activate   # En Windows
```

Paso 2: Configurar CLI de AWS.

- Instalar y conectar la terminal de tu computadora con tu cuenta de AWS. Esto permite enviar comandos desde tu máquina local para crear y administrar servicios en la nube.
- Instrucciones:
- Instalar AWS CLI
 - ➡ Descarga el instalador según tu sistema desde:

 <https://docs.aws.amazon.com/cli/latest/userguide/install-cliv2.html>

- Configurar la conexión

- En la terminal escribe:

```
aws configure
```

- Te pedirá:

- ➔ Access Key ID: Lo obtienes desde tu cuenta de AWS (IAM).
- ➔ Secret Access Key: Igual que el anterior, pero confidencial.
- ➔ Region: Usa `us-east-1` (Virginia) si no estás seguro.
- ➔ Formato de salida: Escribe json.

Paso 3: Preparar el backend (Django) para Elastic Beanstalk.

- Convertir el backend en un proyecto que AWS pueda entender y ejecutar.

- Instrucciones:

- Instalar herramientas necesarias

```
pip install gunicorn  
pip freeze > requirements.txt
```

- Crear archivo `application.py`

- Dentro de `backend/backend`, crea un archivo llamado `application.py` con el siguiente contenido:

```
from backend.wsgi import application
```

- Inicializar Elastic Beanstalk

```
eb init -p python-3.8 backend-aws --region us-east-1
```

- Elige tu configuración y confirma que deseas crear un archivo `elasticbeanstalk`.

- Crear ambiente y desplegar

```
eb create backend-env  
eb open
```

- Esto abrirá tu aplicación Django corriendo en AWS en el navegador.

Paso 4: Preparar el frontend (React) para subir a AWS.

- Empaquetar React para producción y convertir el proyecto React en una página estática lista para subir a internet.

- Instrucciones:

- Crear versión de producción

`npm run build`

- Esto creará una carpeta llamada build con todos los archivos necesarios.

Paso 5: Subir frontend a AWS S3 (o Amplify).

- Usar el servicio S3 o Amplify para mostrar tu frontend como una página accesible por internet.

- Opción A: Con S3 (más manual)

- Crear un bucket

- ➔ Entra a AWS Console > S3 > Crear bucket.
- ➔ Dale un nombre único (ej. `frontend-proyecto-react`).
- ➔ Desactiva el bloqueo de acceso público (deja que sea visible para todos).
- ➔ Da clic en “Crear bucket”.

- Subir archivos

- ➔ Entra al bucket creado.
- ➔ Carga los archivos de la carpeta `build`
- ➔ Habilita la opción de “Hosting de sitios estáticos”.
- ➔ Coloca `index.html` como archivo de inicio.
- ➔ AWS te mostrará una URL para acceder a tu sitio.

- **Opción B: Con Amplify (más automático)**

- ➔ **Ve a AWS Console > Amplify.**
- ➔ **Da clic en “Get started” > “Host web app”.**
- ➔ **Selecciona “Deploy without Git provider”.**
- ➔ **Carga tu carpeta build.**
- ➔ **Amplify subirá y publicará automáticamente tu sitio.**