Prueba 2

Despliegue de una aplicación Django y React.js

Elaborar el deployment dockerizado de una aplicación en django (backend) con frontend en React.js contenida en el repositorio. Es necesario desplegar todos los servicios en un solo docker-compose.

Se deben entregar los Dockerfiles pertinentes para elaborar el despliegue y justificar la forma en la que elabora el deployment (supervisor, scripts, docker-compose, kubernetes, etc)

Subir todo lo elaborado a un repositorio (github, gitlab, bitbucket, etc). En el repositorio se debe incluir el código de la aplicación y un archivo README.md con instrucciones detalladas para compilar y desplegar la aplicación, tanto en una PC local como en la nube (AWS o GCP).

Resolución

Vamos a utilizar Docker para la contenerización de los servicios necesarios:

Backend app - Django - puerto 8000
Frontend app - Node.js - puerto 3000
Database - PostgreSQL

Como primer paso, creamos los Dockerfile respectivos para Backend y Frontend. Este archivo define las instrucciones para buildear y deployar la imagen Docker de nuestras aplicaciones
Definirá: La imagen base a utilizar, dependencias a instalar, código fuente, script entrypoint, entre otros, con todo los necesario que nos permita buildear.

El dockerfile correspondiente a Backend ejecuta como Entrypoint un script personalizado. El cual ejecuta:
Corre script wait-for-it.sh al puerto 5432 (DB), a la espera que la misma se encuentre lista para aplicar los siguientes pasos

Collect static files: Recopila los archivos estáticos para la app, en caso de ser necesario.
Migrate: aplica las migraciones requeridas para que el esquema de la DB se encuentre actualizado
Createsuperuser: Función que permite crear usuario con acceso a los datos sobre los préstamos solicitados.

Posterior procede a el server es iniciado

Crearemos un archivo docker-compose.yml, en el cual definiremos el servicio de la DB, además de administrar, describir y definir interacción entre los distintos servicios
Además para el manejo de variables requeridas, creamos el archivo .env.

Definimos el script init_script.sh, desde el cual podremos hacer todo el set-up del ambiente para el despliegue.
Instalaciones de herramientas, dependencias y código para desplegar nuestra aplicación.

Nos servirá tanto como para hacerlo localmente, o utilizando una instancia EC2

Dentro del código, realizamos algunas modificaciones.

La más significativa fue sobre devops-challenge/backend/apps/request_loan/forms.py donde se define la clase RequestLoanModelForm. Una vez que se ingresan los datos sobre el préstamo a solicitar, estos son enviados y para validación a la API externa http://endpoint.test.com.ar:7001/api/v1/scoring. La misma, y de acuerdo a varias validaciones, actualmente no responde.
Es por eso, y para que permita el funcionamiento correcto de nuestra app, definí:

```
 response_data= {
    'error': False,
    'message': 'Scoring successful',
    'score': random.randint(0, 100)
    }
```

Que nos simulará un response sin errores.