Informe sobre despliegue backend en docker:

paso 1:

Abrimos la terminal de nuestro proyecto y ejecutamos el comando "dotnet restore" para asegurarnos que todos los paquetes nuget esten especificados en el proyecto.

paso 2:

Ejecutamos otro comando "dotnet publish -c Release -o out" Este lo que hace es publicar la aplicacion y manda todo lo publicado a un directorio "out".

paso 3:

Nos dirijimos al directorio y siempre vamos a crear el archivo "Dockerfile" donde se encuentre nuestro archivo de proyecto por asi decirlo nuestra solución. Importante el archivo tiene que ser .txt, pero se le borra el .txt.

paso 4:

Luego nos dirigimos al proyecto y agregamos el archivo que acabamos de crear a nuestro proyecto.

paso 5:

Abrimos nuestro archivo "Dockerfile" y ahi es donde comenzamos a configurar nuestra aplicacion para que se ejecute desde el docker.

paso 6:

Necesitamos descargar la imagen de docker para trabajar con .NET en mi caso que estoy trabajando con .NET 9.0 descargamos esa imagen, tiene que ser la imagen oficial del sdk de .NET "mcr.microsoft.com/dotnet/sdk:9.0" esta imagen la podemos encontrar en docker hub.

paso 7:

primero vamos hacer que docker obtenga la imagen con la palabra clave "From"

FROM mcr.microsoft.com/dotnet/sdk:7.0 AS build

lo que hago en este fragmento de codigo es que docker busque la imagen de .NET SDK y le nombramos build.

paso 8:

luego le vamos a crear un directorio de trabajo a nuestra aplicacion con:

WORKDIR /WebApp

que en mi caso le puse WebApp

paso 9:

luego le asignamos el puerto con:

EXPOSE 5000

En mi caso lo hice con el 5000 que es en el que estoy ejecutando mi proyecto.

paso 10:

luego copiamos el archivo de nuestro proyecto

COPY Web/Web.csproj Web/

con esto estamos que lo copie direntante de mismo directorio.

paso 11:

debajo de copy lo complementamos con:

RUN dotnet restore "web/web.csproj"

que tecnicamente es para verificar que el archivo tenga todas las dependencias necesarias

paso 12:

simplemente vamos a copiar el resto de proyecto:

COPY .
RUN dotnet publish -c Release -o out

Paso 13:

Lo ultimo que vamos a hacer en el codigo es construir la imagen del docker:

FROM mcr.microsoft.com/dotnet/sdk:9.0
volvemos a copiar el FROM por que quiero que la imagen se base en el sdk

WORKDIR /WebApp
COPY --FROM=builder /WebApp/out .
copiamos el from y le decimos que se enfoque en "bouild".

ENTRYPOIT ["dotnet", "web.dll"]
al final creamos el ejecutable.

paso 14:

Abrimos denuevo la termnal y ejecutamos el siguiente comando:

docker -t backend_pqr .

lo que estamos haciendo es crear la imagen.

paso 15:

Luego confirmamos que la imagen se halla creado con docker image con:

docker imagens

ese comando listara las imagenes creadas en el docker.

paso 16:
Luego hacemos el comando para ejecutar el contenedor de nuestra imagen.

docker run -d -p 5000:5000 --nome backenpqr backenpqr

-d = el contenedor se ejecutara en segundoplano, no bloquea la termnal.
-p 5000:5000
el primero es el puerto de la maquina host y el segundo el puerto interno del contenedor

-- name
Es el nombre del contenedor y al final no el nombre de la imagen.