

U5A2 Empaqueta la springboot en Docker

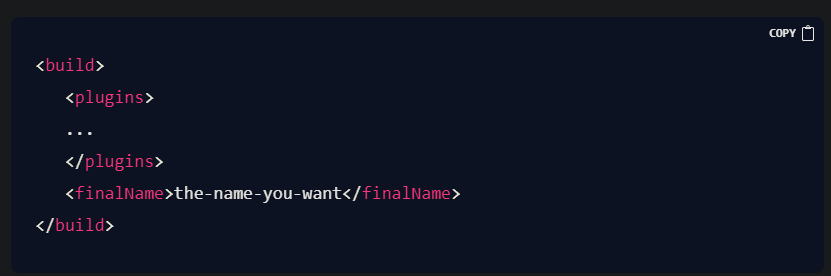
Ismael Bernad Tello 2 ºH

19/11/2023

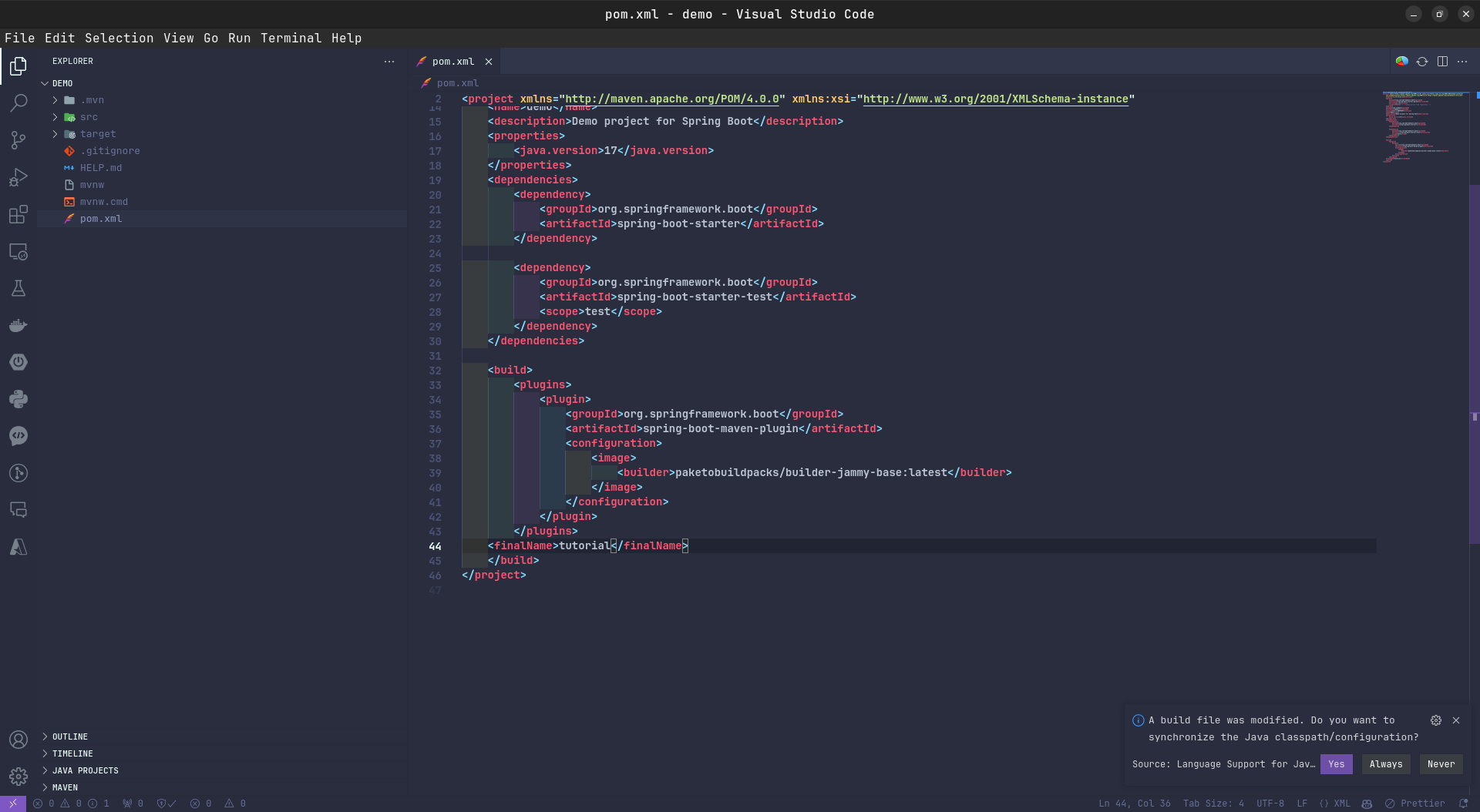
1. Spring Boot + Docker + AWS | Spring Boot and Docker:

**Link al tutorial:** [**https://javaplusstuff.hashnode.dev/java-plus-stuff-series-1-chapter-1**](https://javaplusstuff.hashnode.dev/java-plus-stuff-series-1-chapter-1)

**Primero añadimos lo siguiente al archivo POM.xml del proyecto:**

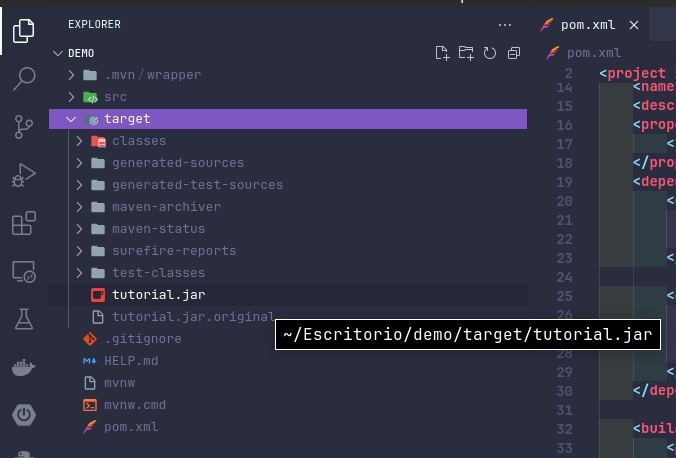
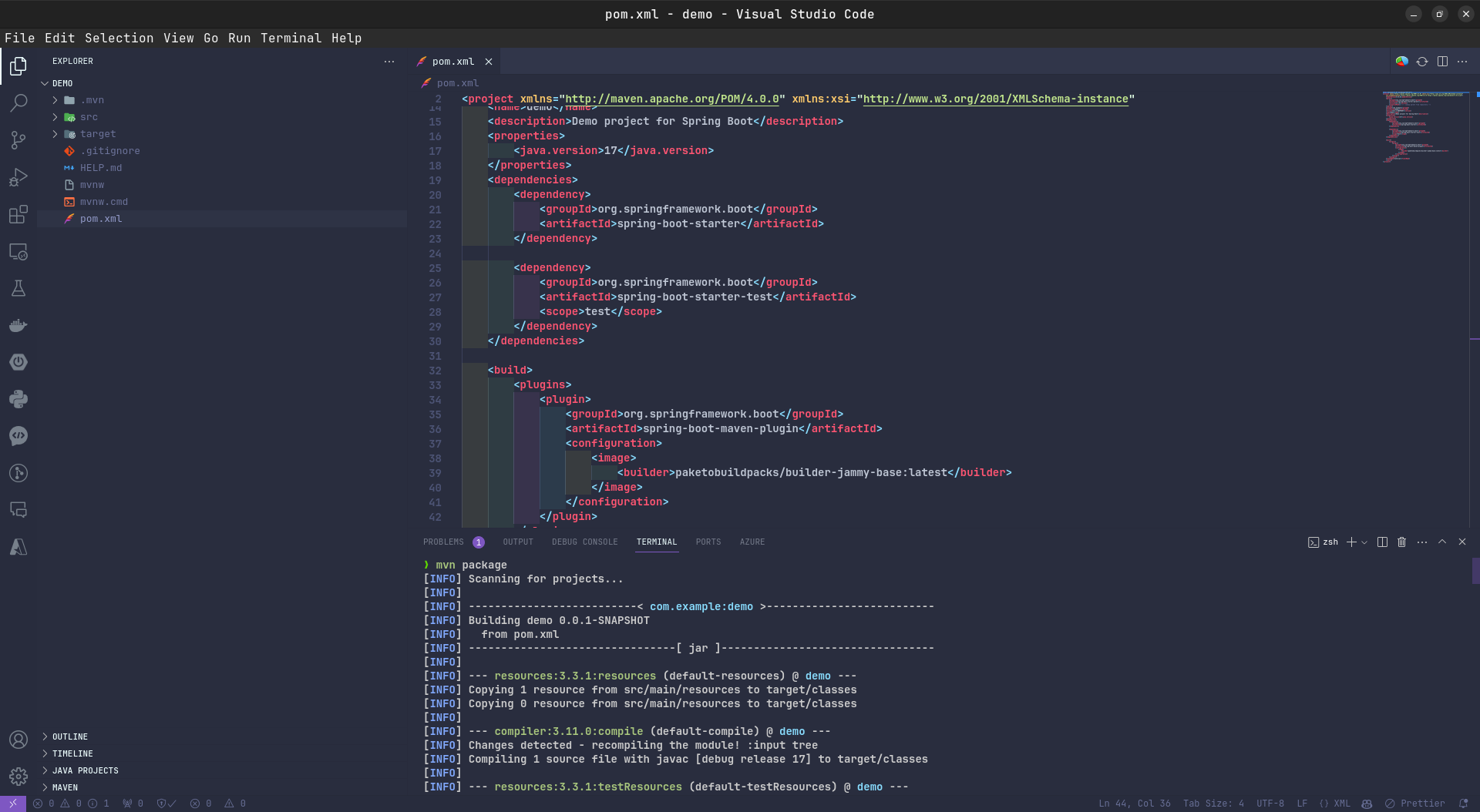


Sustituimos lo que hay dentro de la etiqueta final name por el nombre que queramos dar a el archivo.jar de nuestra aplicación:



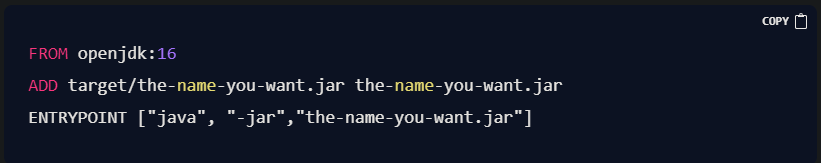
**Creamos un archivo .jar empaquetando nuestro proyecto de springboot :**

Para esto ejecutamos el comando nwn package estando en el directorio donde se encuentra la app de springboot:

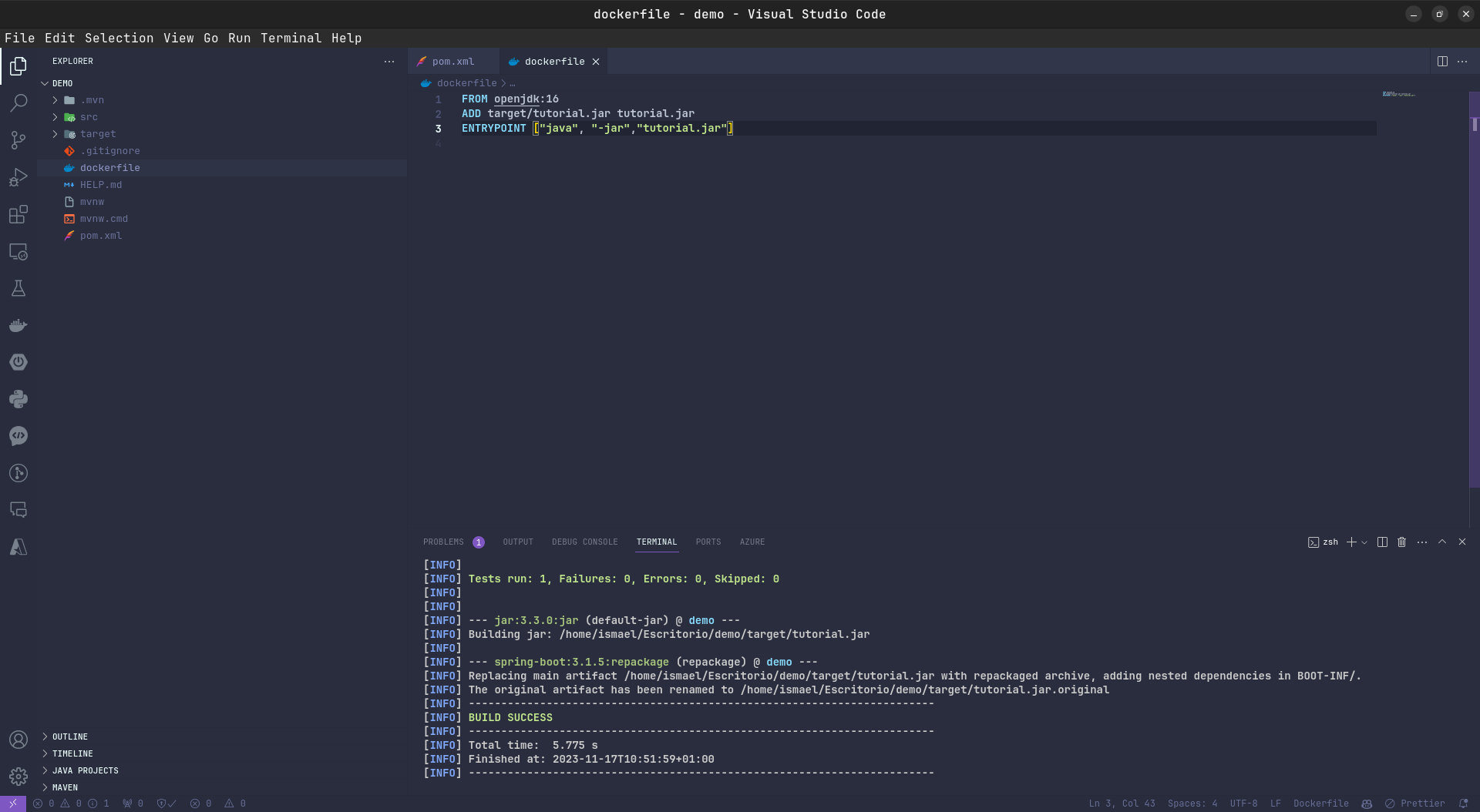


**Dentro del directorio donde se encuentra nuestro proyecto de springboot creamos un archivo llamado Dockerfile:**

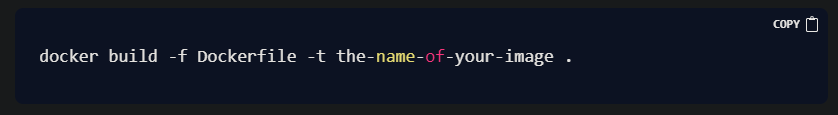
**Dentro del archivo ponemos lo siguiente:**



Modificamos la parte que pone the-name-you-want.jar por el nombre que le hemos dado previamente a la aplicación.jar

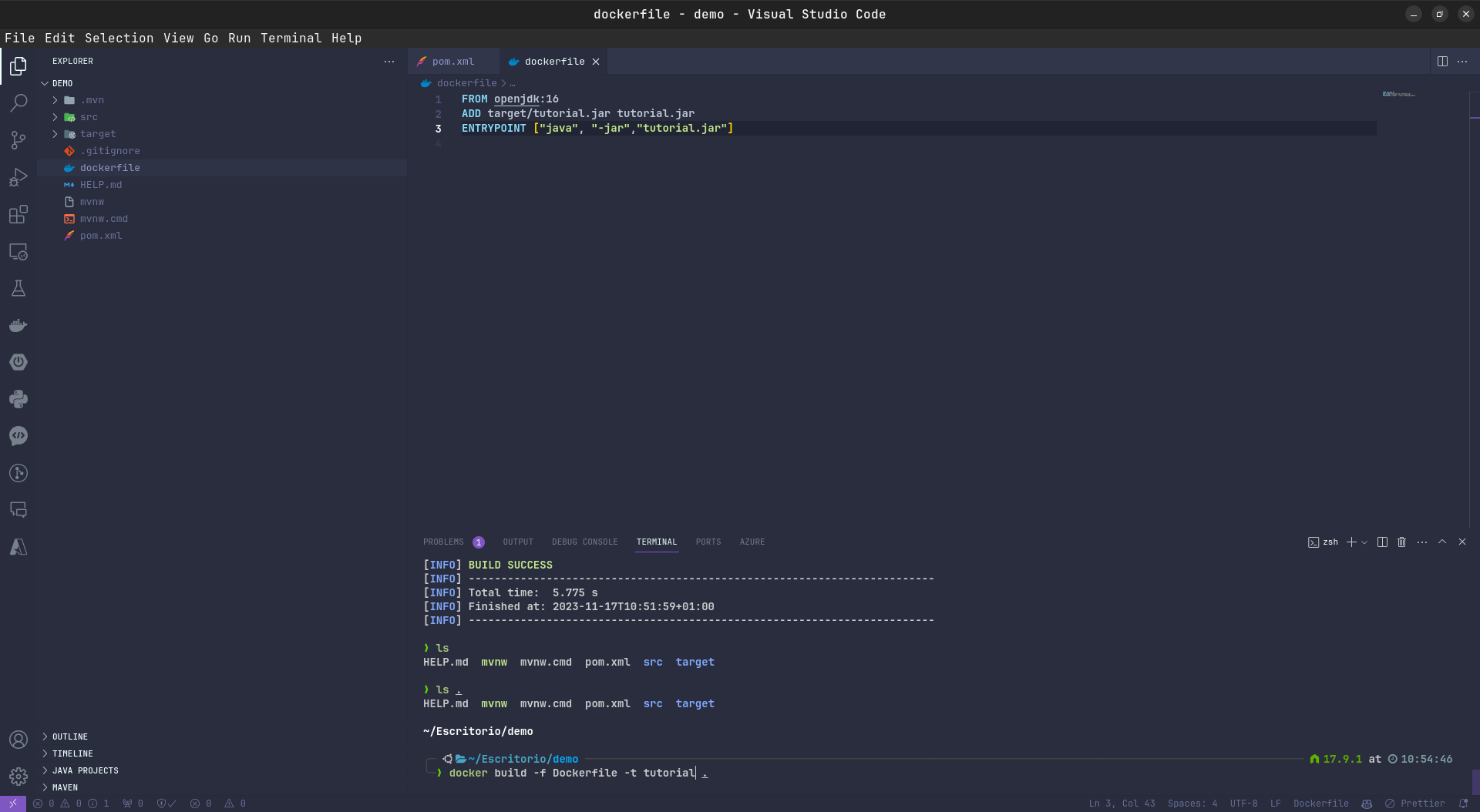


Creamos la imagen de Docker:



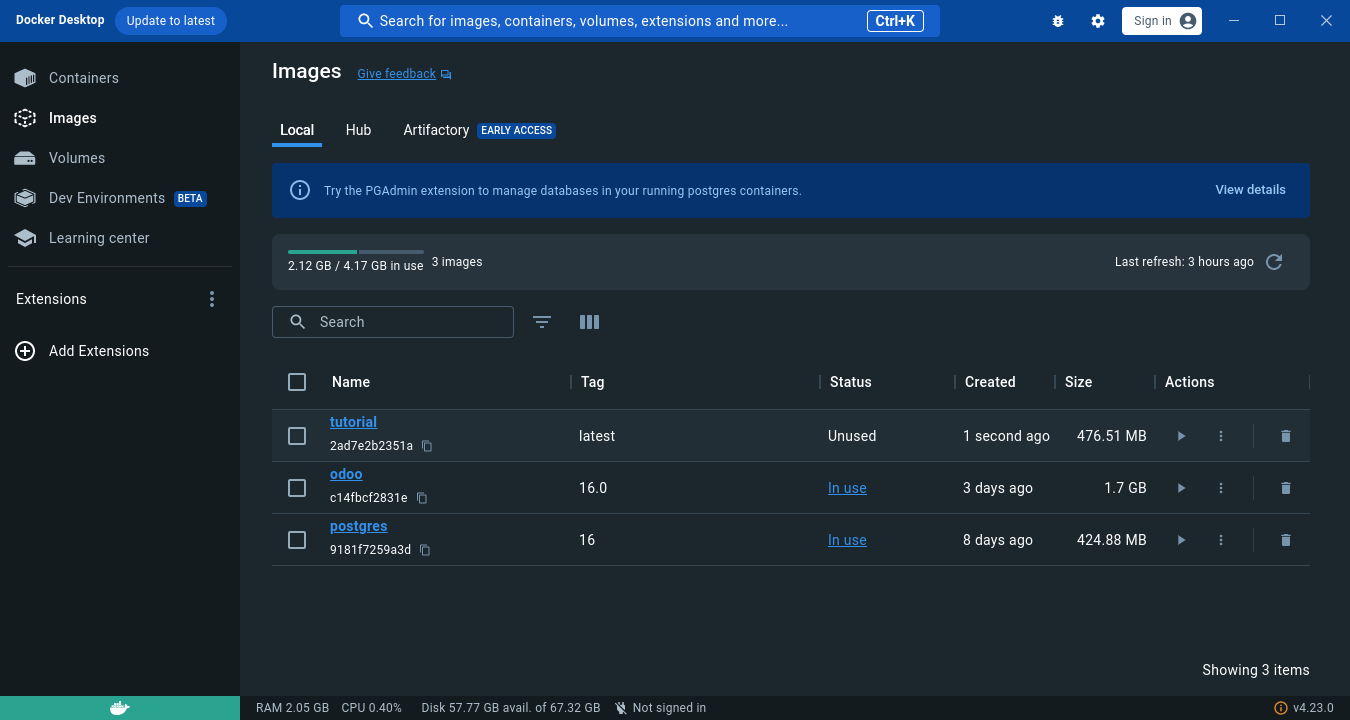
Reemplazamos la parte de the-name-of-your-image . por el nombre de la imagen que queremos.

**\*IMPORTANTE no se borra el punto del final, este es necesario para que funcione. Esto le indica a Docker el punto de inicio para buscar los archivos que necesita para crear la imagen. \***

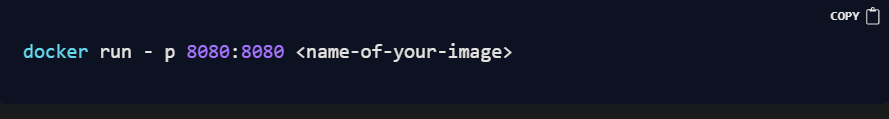


Vemos que la imagen se ha creado correctamente:

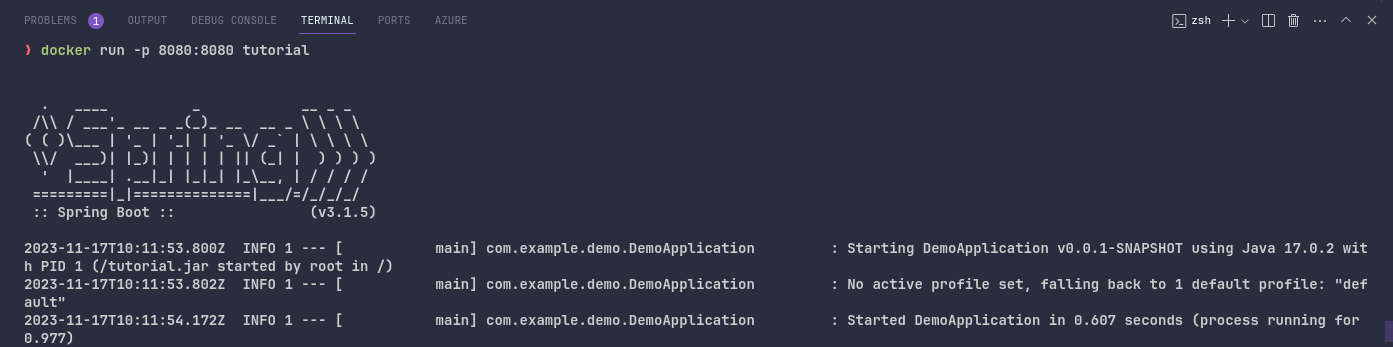
Ejecutamos ‘docker images’ o miramos gráficamente en Docker en el apartado de imágenes:



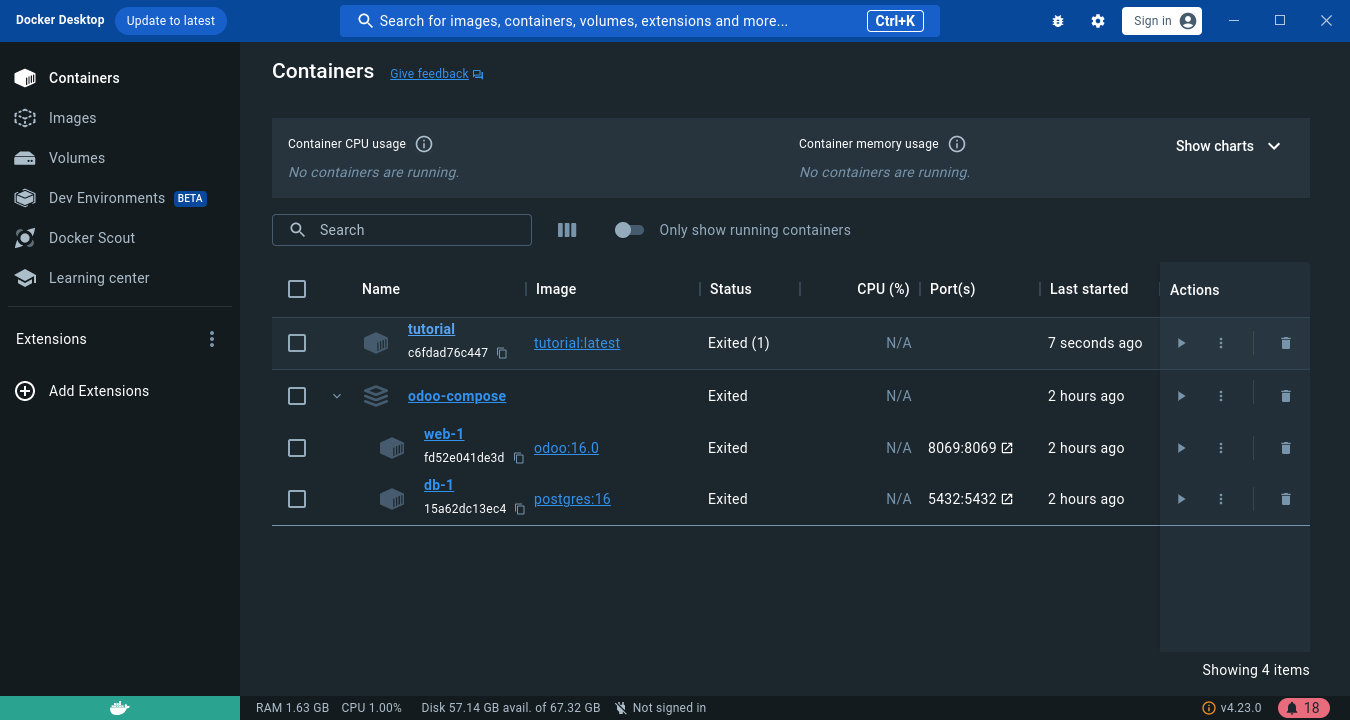
Creamos un contenedor de Docker con la imagen:



Sustituimos el nombre de la imagen por el nombre que le hayamos dado previamente.



Podemos ver que el contenedor se ha creado correctamente:



1. Spring Boot + Docker + AWS | Starting with AWS:

**Link al tutorial:** [**https://javaplusstuff.hashnode.dev/java-plus-stuff-series-1-chapter-2**](https://javaplusstuff.hashnode.dev/java-plus-stuff-series-1-chapter-2)

**Instalamos AWS CLI (Yo ya lo tengo instalado por el comando de Gorka) y saltamos hasta esta parte del tutorial.**

**Configurar la conexión a AWS:**

Nos metemos dentro de un laboratorio AWS y buscamos las credenciales de AWS CLI.

**CAPTURA\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

Copiamos el key ID y el Access key.

**Nos dirigimos al terminal y escribimos ‘aws configure’ e insertamos el key ID y Access Key cuando se nos pregunte:**

**CAPTURA\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Podemos ver que nos hemos conectado correctamente a la consola del laboratorio:**

**CAPTURA\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**El tutorial termina aquí y no se puede hacer nada más debido a los permisos y servicios que tienen las cuentas educativas.**

**Si quieres continuar el tutorial puedes hacer la parte 3 y 4 del mismo:** [**https://javaplusstuff.hashnode.dev/series/spring-boot-plus-aws?authuser=0**](https://javaplusstuff.hashnode.dev/series/spring-boot-plus-aws?authuser=0)