**Consigna:**

Crear un contenedor para la aplicación PasswordApi (del ejercicio 4) que esté basada en la imagen redhat-openjdk-18/openjdk18-openshiftdisponible en la registry de Red Hat (<https://catalog.redhat.com/software/containers/redhat-openjdk-18/openjdk18-openshift/58ada5701fbe981673cd6b10>).1. Escribe el dockerfile2. Genera la imagen (docker build)3. Publica la imagen en tu cuenta de DockerhubPon el dockerfile en un repositorio Github en una carpeta ejercicio6. Agregar también en esa carpeta un archivo readme con el link a la imagen generada en dockerhub.Entrega el link al repositorio Github.

**RESOLUCIÓN:**

1. **Obtener la imagen de redhat**

Para ello creo una cuenta en redhat.

Con la credenciales generadas me logueo en el registry de redhat y descargo la imagen del registry mencionado.

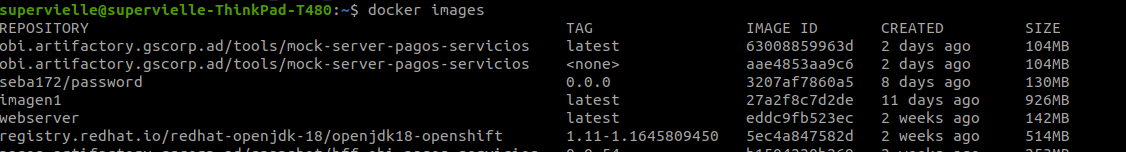
docker login registry.redhat.io

Username: seba172

Password: XXXXXXXXX

Login Succeeded!

docker pull registry.redhat.io/redhat-openjdk-18/openjdk18-openshift:1.11-1.1645809450

Verifico que se descargo la imagen y la tengo en mi registry local, a partir de este paso puedo generar el dockerfile.

leo la documentacion

en teoria

-el working directory figura en la doc es "/home/boss"

-no hace falta agregar el user porque la imagen ya lo tiene

-los puertos tambien los trae expuestos por lo que tampoco hace falta configurarlo

1. **Se crea el dockerfile**

FROM registry.redhat.io/redhat-openjdk-18/openjdk18-openshift:1.11-1.1645809450

COPY passwordapi.jar /home/jboss/app.jar

ENTRYPOINT ["java", "-jar", "/home/jboss/app.jar"]

FROM registry.redhat.io/redhat-openjdk-18/openjdk18-openshift:1.11-1.1645809450

Indica que queremos usar la versión de redhat.

COPY passwordapi.jar /home/jboss/app.jar

Indicamos la ubicación de nuestro jar de aplicación y a donde lo vamos a copiar en el contenedor. En este caso la imagen nos indica que el directorio destino es home/jboss

ENTRYPOINT ["java", "-jar", "/home/jboss/app.jar"]

Se indica el comando de arranque que debe utilizar el contenedor una vez inicializado.

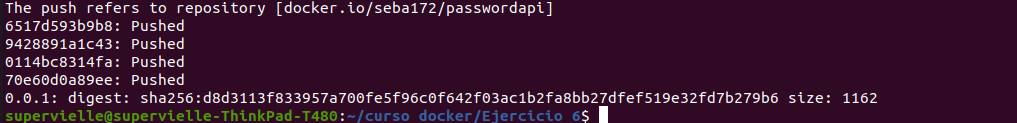
1. **Se genera la imagen a partir del dockerfile del punto 1**

**Explicación:**

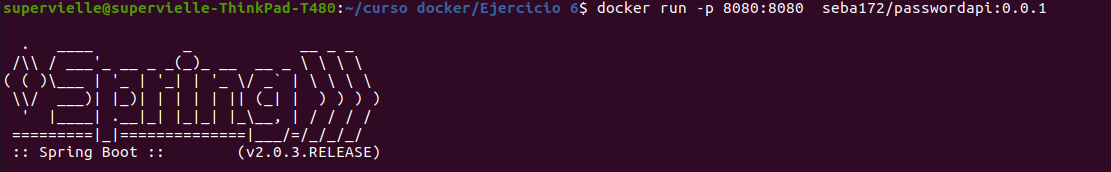
**docker build -t my\_docker\_hub\_username/my\_image\_name:my\_image\_version .**

Importante el punto (.) al final, este indica el directorio actual, en donde está el fichero Dockerfile. el argumento -t user/image:tag permite nombrar la imagen

1. **Se sube al repositorio de dockerhub la imagen generada**

****

1. **Se corre la imagen mapeando en los puertos 8080**

****

**Se verifica que está funcionando**

