**Docker Desktop**

Algunas funciones de Docker Desktop se utilizarán para ayudarnos a ejecutar nuestros contenedores y administrar nuestras imágenes. Este es un ejemplo de cómo encajar una aplicación en Java.

Paso 1: Cree nuestro código o aplicación en Java, para que podamos implementar dicho código, podemos usar la herramienta Spring Boot, en esta demostración usaremos Maven, la versión de Java que queremos, definiremos la forma del paquete y finalmente la dependencia, en este caso web dependencia, luego haga clic en el botón Crear.

Paso 2: abra el código de Visual Studio, para ver el código descargado de Spring Boot, para poder administrar las dependencias, una vez que importemos RequestMappin y RestController, podemos crear un método de respuesta para nosotros usando la URL de la API y responder a nosotros con un "hola"

**COMPILAR CODIGO**

Creamos nuestro jar con el paquete ./mvnw, luego miramos el destino para verificar que se creó y luego leemos la compilación de Dockerfile donde enviamos un argumento de que el jar de correo está destinado donde lo que hace es copiar el jar al contenedor y ejecutar la instrucción ENTRYPOINT.

A continuación ejecutamos el comando java -jar target/spring-boot-docker-0.0.1-SNAPSHOT.jar donde este empieza a iniciar una aplicación en springboot. Para saber por cual puerto se inicia debemos mirar en el Tomcat started on ports: ###, con el propósito de verificar de que es el servicio correcto y que está funcionando.

**CREAR UNA IMAGEN**

Creamos una imagen a través de DockerFile, usando el comando Docker build -f Dockerfile-build -t imageName. Con esto comenzará a construir la imagen, para verificar que esta imagen existe usamos el comando Docker image ls, luego ejecutamos la imagen generada en un contenedor con el comando Docker run -p 8080:8080 -t spring-boot con esto ejecute spring desde el contenedor, usando el comando Docker container ls podemos ver que el contenedor se está ejecutando.

**ABRIR DOCKER**

En Docker podemos ver sus variables de entorno, dónde está java, sus argumentos y el comando de ejecución, donde podemos ejecutar un contenedor y darle el nombre que queramos, como: my\_first\_container\_spring\_boot y establecer puertos y tamaños. El contenedor ahora ejecuta nuestra imagen Springboot, que nos permite ver lo que se está ejecutando donde podemos ir al navegador y ejecutar localhost 8080 y responderá desde el contenedor con "hola palabra", donde podemos ver todo tipo de cosas. contiene, como sus estadísticas; También podemos detener nuestro contenedor, ya que dirá que no puede responder; de lo contrario, la aplicación responderá. Podemos reiniciar nuestro contenedor, si falla al final, podemos eliminarlo.