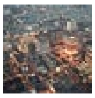
Build a Docker Image using Maven and Spring boot



Ramanamuttana

.

Follow

4 min read

.

Sep 12, 2023

--

2

Listen

Share

#1. Through Docker file

To build a Docker image using Maven and Spring Boot, you can follow these steps:

Create a new Maven project with Spring Boot by using the Spring Initializr or by adding the necessary dependencies to an existing Maven project.

Add a Dockerfile to the root directory of your project. The Dockerfile is a text file that contains instructions for building the Docker image. Here's an example of a Dockerfile for a Spring Boot application:

```
# Use an official Maven image as the base image
FROM maven:3.8.4-openjdk-11-slim AS build
# Set the working directory in the container
WORKDIR /app
# Copy the pom.xml and the project files to the container
COPY pom.xml .
COPY src ./src
# Build the application using Maven
RUN mvn clean package -DskipTests
# Use an official OpenJDK image as the base image
FROM openjdk:11-jre-slim
# Set the working directory in the container
WORKDIR /app
# Copy the built JAR file from the previous stage to the container
COPY - from=build /app/target/my-application.jar .
# Set the command to run the application
CMD ["java", "-jar", "my-application.jar"]
```

Open a terminal or command prompt and navigate to the root directory of your project.Build the Docker image by running the following command:

```
docker build -t my-application .
```

This command builds the Docker image using the Dockerfile in the current directory and tags it with the name "my-application".

Once the Docker image is built, you can run it using the following command:

```
docker run -p 8080:8080 my-application
```

This command runs the Docker container and maps port 8080 of the container to port 8080 of the host machine.

Your Spring Boot application should now be running inside a Docker container. You can access it by opening a web browser and navigating to http://localhost:8080.

#2 .Using the integrated Spring Boot build-image goal

NOTE: Make sure docker desktop is running in windows before creating an image .

You don’t need to make any changes because Spring Boot already has a plugin for creating Docker images because it is accessible through the spring-boot-starter-parent standard that is included in your pom.xml.

Additionally, you do not need to create a Dockerfile because the plugin handles the security, memory, and performance enhancements that Spring advises. However, heed this warning:

NOTE:if your Dockerfile is located in your source code repository, it will be disregarded or ignored.

Consequently, run the command without making any other changes to the code.Open the Terminal

View →**Tool Windows** →**Terminal** and type the following command

```
mvn spring-boot:build-image
```

type in Terminal spring boot app directory

When everything is finished, a notification should appear stating that the Docker image was successfully built , names are taken from the groupId or from name and version included

image name taken before from artifact id
Image created, a notification on Terminal

Now ,we can see the image in docker desktop

image from docker desktop

change image name

incase if you want to change the name of the image other than the mentioned artifactid or name than you can specify the name of the image in the pom.xml as shown in below code , for Ex:sampleimage and run with the above mentioned command .

```
 <build>
       <plugins>
           <plugin>
               <groupId>org.springframework.boot</groupId>
               <artifactId>spring-boot-maven-plugin</artifactId>
               <configuration>
                   <image>
                       <name>sampleimage</name>
                   </image>
               </configuration>
           </plugin>
       </plugins>
   </build>
```

docker image with preferred name

Note : Make sure your docker desktop is not running in **admin** mode , incase if it is running in admin mode only ,change it by add the user to docker group by typing the following command on command prompt which run as an ***admin mode*** .

```
net localgroup docker-users "your-user-id" /ADD
```

your-user-id is generally your windows username