Create docker container spring boot application with maven and docker compose

Serafeim Kourlos

.

Follow

Published in

The Startup

.

3 min read

.

Jun 24, 2020

--

Listen

Share

In this tutorial we will check how to make a java spring boot with maven in a docker container with docker compose.

There are some reasons to have a setup like this. The obvious reason is if you don not have and you do not want to install maven in your local machine. Another reason can be that you work for a client and all the repos in your settings.xml are pointing to that client, so if you want to run your personal projects you need to configure again the settings.xml. I personally use it of the ease of use and the flexibility that it has such a setup.

You can clone the entire project in the following github repository (https://github.com/sera13/docker-demo.git)

Let's see what this repository contains. First of all the structure of the project looks like below

Structure of the project

DockerFile + entrypoint.sh

As it can be seen in the Dockerfile we just use the base image from the maven docker-hub repository and we copy inside it the entrypoint.sh in order to execute it when a new container is instantiated. The reason that I put the entrypoint.sh and not just directly put **'CMD mvn spring-boot:run'** is that in a real scenario normally you need to do some things to initialize an application server like copy folders, execute other scripts, set variables, check external server availability etc. That's why it is more flexible and maintainable to write an entrypoint.sh file.

Just to avoid unpleasant surprises like the error below (this error occurs when we commit entrypoint.sh with Windows line separators [CRLF])

```
standard_init_linux.go:175: exec user process caused "no such file or directory"
```

we need to include following in the Dockerfile

```
RUN apt-get update && apt-get install dos2unix && dos2unix /usr/local/bin/entrypoint.sh && chmod +x /usr/local/bin/entrypoint.
```

Docker compose + .env file

As it can be seen from the .env file, I chose to use my local machine's maven settings. For this to work I just need to mount the .m2 folder from my local machine. This way I avoid every time that we create a new container to download all the maven dependencies again and again. However if you do not want to install maven in your local machine you just need to comment the 'MAVEN_SETTINGS_FOLDER' in the .env file and in docker-compose.yml.

pom.xml

The pom file is just basic spring boot application. The only special thing is

```
Xrunjdwp:transport=dt_socket,server=y,suspend=n,address=*:5005
```

which is used in order to remotely debug your project in port 5005.

Run the project

Open a command line and run the following commands

```
git clone https://github.com/sera13/docker-demo.git
cd docker-demo
```

Adjust the .env file according to your local file system and then you have 2 options.

If you want to run the project with docker then in the docker-demo folder run

```
docker-compose up
```

if you want to run directly with maven then

```
mvn clean install spring-boot:run
```

Then go to http://localhost:8080/ and check if the server is up.

Conclusion

I have tried to keep this project simple for anyone in order to run an initial setup of spring boot with docker, docker-compose, .env and entrypoint.sh. If you cannot set it up just ask for my help. Please let me know if it was helpful for you.