

Title: Execute mvn clean install before docker-compose commands

Post Body:

I am running multiple spring-boot microservice within docker containers. I am writing docker-compose.yml file to automate de deploy, but I have a problem. All the spring-boot microservices have different profiles depending on if you want to run them locally without docker and another ig you want to run them with docker (Basically each profiles changes the URIs of clients between microservices).

If I run docker-compose up and the jar files that have been compiled are not compiled with 'docker' spring-boot profile, it will creates the images with the wrong jar.

MY DOCKER-COMPOSE.YML EXAMPLE:

```
version: '3'  services:    imageserver:      build:          context: ./ImageServer      dockerfile: Dockerfile    ports:  -
```

ONE OF MY DOCKERFILE EXAMPLE:

```
FROM openjdk:8-jdk-alpine LABEL maintainer='lalala@gmail.com' VOLUME /tmp EXPOSE 8082 COPY ./target/*.jar Core.jar ENTRYPOINT
```

is there any way to execute a command like 'mvn clean install -P docker' in each directory before running docker-compose to ensure that the jar that will be included in the image has been compiled with the right profile? The command must be excited BEFORE cretae the container (out of the container to complile the jar that will be included within the container NOT inside the container)

Thanks

Accepted Answer:

I found the better solution! <http://www.littlebigextra.com/use-spring-profiles-docker-containers/>

I can compile without usisng a profile but then i can use java -jar selecting the spring profile!

thanks for your comments, responses!

Highest Rated Answer:

You must be able to use RUN on your Dockerfile, and give `maven clean install`. But why do you want to do that? Integrate Docker build into your maven, and configure the same on your profile where you want to build the images. [Refer here](#).