

Title: how do i store maven dependencies inside docker image using a Dockerfile

Post Body:

so im making a spring boot application that im supposed to put in and run from a docker container, and i would like to build the whole image using a docker file. im using this dockerFile:

```
FROM openjdk:8-jdk-alpine ADD . /analytics-service WORKDIR /analytics-service ENTRYPOINT ./mvnw spring-boot:run
```

when i create the image it just copies the files, and only after i run it, it starts downloading all the maven dependencies. which takes a while, given that i will be running a few containers. so how do i do it ? i want it to get all the dependencies when the image is created, so when i create a container it doesnt start downloading.

Accepted Answer:

If I understood you correctly, you would like to have the Maven dependencies downloaded first then combine them with your app into an image?

If that's what you want then the proper way to do that is as follows:

1. Pull a maven image (normally you call this stage the 'builder' - it's just a name)
2. Copy your pom.xml file into a working directory
3. Run maven once to get your dependencies and then again to package it up
4. Create a new image base from openjdk
5. Copy the result of step 3 into your app image
6. Expose a port
7. Provide an entry point

Here's what that looks like in a Dockerfile:

```
FROM maven AS builder WORKDIR /usr/src/analytics COPY pom.xml . RUN mvn -B dependency:go-offline COPY . . RUN mvn package FR
```

You'll need to know how your jar file is named before you run this. You can run mvn package outside of Docker on your computer and see the filename that is generated. Copy that into the two spots above in the Dockerfile.

Highest Rated Answer: None