

Title: Do not download all Maven dependencies on a Docker build

Post Body:

I'm trying to create a Dockerfile to then build a Maven project.

I wonder how to fix the Dockerfile and what command to then execute.

I would like to know how to run the build so that it does NOT download all the Maven dependencies every time it builds when the source code, sitting in the src/ directory, has NOT changed.

Here is my Dockerfile file:

```
FROM maven:3.3.9-jdk-8 RUN mkdir -p /usr/src/app WORKDIR /usr/src/app RUN cd /usr/src/app ADD pom.xml /usr/src/app RUN mv
```

Should I run the `docker run --rm -it toolbox` command or the `docker build -t toolbox .` command ?

Both of these above commands run fine, except that they both download all the Maven dependencies even if the source code has not been touched.

Accepted Answer:

That's how Docker works. Every time you do `docker run`, it creates a new container which does not have any access to the files in the old container. So, it download all dependencies it requires. You can circumvent this by declaring an external volume. Looking at the `Dockerfile` of Maven, it declares a volume `/root/.m2`. So, you can use a directory in your host machine and attach it to this volume by `-v` option. Your Docker command would be,

```
`docker run -v <directory-in-your-host>:/root/.m2 <other-options-and-commands>
```

Every time you run a new `docker run`, Maven will look into your local directory before downloading the dependency.

However, my question is why don't you build your app first and use the resulting jar to create the docker images unless you have any specific reasons. You can create your own Dockerfile using java base image or simply use one of the `docker-maven-plugin` like [spotify](#) available out there. That makes your life a lot easier.

Highest Rated Answer: None