

Title: Maven docker cache dependencies

Post Body:

I'm trying to use docker to automate maven builds. The project I want to build takes nearly 20 minutes to download all the dependencies, so I tried to build a docker image that would cache these dependencies, but it doesn't seem to save it. My Dockerfile is

```
FROM maven:alpine RUN mkdir -p /usr/src/app WORKDIR /usr/src/app ADD pom.xml /usr/src/app RUN mvn dependency:go-offline
```

The image builds, and it does download everything. However, the resulting image is the same size as the base `maven:alpine` image, so it doesn't seem to have cached the dependencies in the image. When I try to use the image to `mvn compile` it goes through the full 20 minutes of redownloading everything.

Is it possible to build a maven image that caches my dependencies so they don't have to download everytime I use the image to perform a build?

I'm running the following commands:

```
docker build -t my-maven . docker run -it --rm --name my-maven-project -v '$PWD':/usr/src/mymaven -w /usr/src/mymaven my-maven mvn compile
```

My understanding is that whatever `RUN` does during the docker build process becomes part of the resulting image.

Accepted Answer: None

Highest Rated Answer:

Usually, there's no change in `pom.xml` file but just some other source code changes when you're attempting to start docker image build. In such circumstance you can do this:

FYI:

```
FROM maven:3-jdk-8 ENV HOME=/home/usr/app RUN mkdir -p $HOME WORKDIR $HOME # 1. add pom.xml only here ADD pom.xml $HOME
```

So the key is:

add `pom.xml` file.

then `mvn verify --fail-never` it, it will download maven dependencies.

add all your source file then, and start your compilation(`mvn package`).

When there are changes in your `pom.xml` file or you are running this script for the first time, docker will do 1 -> 2 -> 3. When there are no changes in `pom.xml` file, docker will skip step 1 and do 3 directly.

This simple trick can be used in many other package management circumstances(`gradle`■`yarn`■`npm`■`pip`).

Edit:

You should also consider using `mvn dependency:resolve` or `mvn dependency:go-offline` accordingly as other comments & answers suggest.