

Title: How to use maven-docker on development-environment to avoid repeated download jars?

Post Body:

this is my Dockerfile,but if I changed the java code ,and then enter 'docker-compose up -d --build' to terminal,maven will re-download the jars, how to avoid duplication of the download?

Dockerfile:

```
FROM maven:3.5.0-jdk-8 WORKDIR /app COPY pom.xml . COPY src ./src RUN mvn verify ENTRYPOINT ['java','-jar','target/cloud-user.
```

docker-compose.yml:

```
version: '2' services:  cloud-nginx:    build: cloud-nginx    image: registry.cn-beijing.aliyuncs.com/wangdali/cloud-nginx
```

I streamlined my test project, and then use the spring-boot plugin for maven

```
mvn spring-boot:run
```

to start the project, now that mount .m2 folder first and then run mvn command. But after I changing the java source code ,it does not loading automatically by devtools, how to solve?

and this is my Dockerfile/docker-compose.yml :

Dockerfile:

```
version: '2' services:  user:    build: user    volumes:    - $HOME/.m2/repository:/root/.m2/repository    - ./user/s
```

docker-compose.yml:

```
FROM maven:3.5.0-jdk-8 WORKDIR /app COPY pom.xml . COPY src ./src ENTRYPOINT ['mvn','spring-boot:run']
```

Accepted Answer:

Resolved, I mount target folder to container,it's can be automatically loaded, which is the final dockerfile and docker-compose.yml, of course, in the production environment, we can not write:

Dockerfile:

```
FROM maven:3.5.0-jdk-8 WORKDIR /app COPY pom.xml . COPY src ./src ENTRYPOINT ['mvn','spring-boot:run']
```

docker-compose.yml

```
version: '2' services:  user:    build: user    volumes:    - $HOME/.m2/repository:/root/.m2/repository    - ./user/t
```

pom.xml:

```
<plugins>    <plugin>    <groupId>org.springframework.boot</groupId>    <artifactId>spring-boot-maven-plugin</artifactId>
```

Highest Rated Answer: None