

Title: Dockerfile can't build app with Maven - No goals specified for this build

Post Body:

I'm developing a Spring Boot app with several data services, but I'm struggling to get it working as a containerized application. This is the project folder structure:

```
myapp/
├── src/
│   ├── ...
│   ├── docker-compose.yml
│   ├── Dockerfile
│   ├── init-mongo.js
│   └── pom.xml
```

And this is the Dockerfile content I came up with after reading tutorials and official documentation:

```
# # Build stage #
FROM maven:3.5-jdk-8 AS build
COPY init-mongo.js /docker-entrypoint-initdb.d/
COPY src /app/src/
COPY pom.xml /app/pom.xml
```

And this is my docker-compose.yml:

```
version: '3'
services:
  app-mongodb:
    container_name: app-mongodb
    image: mongo
    environment:
      MONGO_INITDB_DATABASE: myapp_db
```

Thing is *it should* work, but I keep getting Maven error whenever I run the `docker-compose up` command:

```
myapp_1 | [INFO] Scanning for projects... myapp_1 | [INFO] -----
```

What am I missing?

EDIT: I tried explicitly packaging the JAR file and then just copying it to the end container editing the Dockerfile content, as suggested by @BeppeC:

```
FROM openjdk:8-alpine
COPY target/*.jar /usr/app/app.jar
ENTRYPOINT ["java", "-jar", "/usr/app/app.jar"]
```

It's still producing the same error, so it seems there's something triggering the `mvn build` which I'm not aware of.

EDIT2: I suspect what is causing the "no goals" error might be something wrong with my `pom.xml` file:

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.example</groupId>
  <artifactId>myapp</artifactId>
  <version>1.0.0</version>
  <packaging>jar</packaging>
  <name>MyApp</name>
  <description>MyApp</description>
  <url>http://example.com</url>
  <inceptionYear>2017</inceptionYear>
  <parent>
    <groupId>com.example</groupId>
    <artifactId>parent</artifactId>
    <version>1.0.0</version>
  </parent>
  <properties>
    <java.version>1.8</java.version>
  </properties>
  <dependencies>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter</artifactId>
      <version>2.0.0</version>
    </dependency>
  </dependencies>
  <build>
    <plugins>
      <plugin>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-maven-plugin</artifactId>
      </plugin>
    </plugins>
  </build>
</project>
```

It seems alright to me, but again I'm working my way through tutorials so I might've missed something important.

Accepted Answer: None

Highest Rated Answer:

Works for me if I change from:

```
# # Build stage #
FROM maven:3.5-jdk-8 AS build
COPY init-mongo.js /docker-entrypoint-initdb.d/
COPY src /app/src/
COPY pom.xml /app/pom.xml
```

TO:

```
# # Build stage #
FROM maven:3.5-jdk-8 AS build
COPY init-mongo.js /docker-entrypoint-initdb.d/
COPY src /app/src/
COPY pom.xml /app/pom.xml
```

Main problem I saw was purely that no `pom` was available in the current working dir.. / (root dir).. setting the `WORKDIR` to the `/app` worked perfectly though.