

Title: .jar file not found when building a Docker container with Palantir Gradle plug-in

Post Body:

If I try to build a Docker container with a Spring Boot application under Windows 10, I get the following error:

```
> Task :docker FAILED COPY failed: stat /var/lib/docker/tmp/docker-builder711841135/myproject.jar: no such file or directory
```

I'm using Docker Community Edition in version 18.03.0-ce-win59 (16762) and Gradle 4.7 with Java 8.

**build.gradle** (shortened):

```
plugins {
    id 'java'
    id 'org.springframework.boot' version '2.0.1.RELEASE'
    id 'com.palantir.docker' version '0.19.2'
```

**Dockerfile** (sibling of build.gradle in the top-level project directory):

```
FROM openjdk:8-jre COPY build/libs/myproject.jar myproject.jar ENTRYPOINT ['java', '-Djava.security.egd=file:/dev/./urandom', '']
```

If I build the Docker container with Docker only (without Gradle) it works.

How can I let Gradle (or Docker?) find the file myproject.jar?

Accepted Answer:

The problem is the COPY command in the Docker file:

```
COPY build/libs/myproject.jar myproject.jar
```

The source directory build/libs/ is not where the files for building the Docker container reside. Instead the directory build/docker/ is used as Docker build context. When COPY is executed this directory is the effective working directory.

The correct COPY command is as simple as this:

```
COPY myproject.jar /
```

Docker task:

```
docker {
    dependsOn bootJar
    name '${project.group}/${jar.baseName}:${version}'
    files bootJar.archivePath }
```

If you want to copy resources too, you need to add processResources to the files parameter:

```
files bootJar.archivePath, processResources
```

Highest Rated Answer:

It works with following config of the docker task:

```
docker {
    dependsOn build
    name '${project.group}/${jar.baseName}'
    files bootJar.archivePath
    buildArgs(['JAR_FILE'
```

Using this slightly different docker file:

```
FROM openjdk:8-jre ARG JAR_FILE COPY ${JAR_FILE} app.jar ENTRYPOINT ['java', '-Djava.security.egd=file:/dev/./urandom', '-jar', '']
```