Title: docker-compose java cannot access jarfile

Post Body:

I'm trying to run a Spring Boot app as one of the services in a Docker Compose setup, but whenever I run `docker-compose up -d` the container running the Spring Boot app fails and exits immediately. For purposes of this post I'm using generic names like 'mycontainer' and 'myfile'.

Running `docker-compose logs core` shows that the error message is:

```
mycontainer | Error: unable to access jarfile myjar.jar
```

My docker-compose file is:

```
version: '3.3'  services:   webapp:     image: webapp_image     container_name: webapp-container     build:       context: ./w
```

The Dockerfile for the `core` service, which is the one running the Spring Boot app, is this:

```
FROM openjdk:8-jdk-alpine RUN mkdir -p /usr/src/core RUN mkdir -p /usr/src/algorithms ADD ./myproject/target/myjar.jar /usr/sr
```

If I run an individual container from the `core_image` image, and set the same environment vars that I defined in the docker-compose file, I can launch the Spring Boot app successfully, like so:

```
$ docker run -it -p 8080:8080 --name mycontainer core_image sh $ /usr/src/core # ls -l total 15852 -rwxr-xr-x    1 root     ro
```

But if I run a container from the same image as part of `docker-compose up -d` the container fails with the error that Java couldn't access the jarfile. I've tried specifying the absolute path to the jarfile in the `CMD` of the Dockerfile, but that doesn't help when running from Docker Compose.

Does anyone know what might be causing this? I'm on a Windows 10 host. Any help is greatly appreciated.

Accepted Answer:

It seems you are overriding content of /usr/src/core [where you copy the jar file to] by mounting content from ./core in docker-compose. Remove below mount from services->core->volumes and give a try. `volumes: - './core:/usr/src/core'`

Highest Rated Answer: None