

Title: Docker multi-stage build fails if we use CMD in dockerfile

Post Body:

Dockerfile:

```
FROM maven:3.6.3-openjdk-8 as builder # Set the working directory. WORKDIR /usr/src/mymaven COPY ./ /usr/src/mymaven CMD [ "mv
```

The above docker build fails with the error: target folder does not exist

The below dockerfile works perfectly:

```
FROM maven:3.6.3-openjdk-8 as builder # Set the working directory. WORKDIR /usr/src/mymaven COPY ./ /usr/src/mymaven RUN [ "mv
```

Just changing CMD to RUN fixed the issue. Why is this happening? I thought in the intermittent container, the cmd would execute which should make both the commands equivalent right?

Accepted Answer:

In a multistage build, you may copy files from a previous step. Each step is considered as an individual, private **image**(in the scope of the multistage build).

CMD instruction however is not invoked at build time, it only applies at runtime as clearly stated in the [official docs](#):

The main purpose of a CMD is to provide defaults for an **executing container**.

Since you are currently building the result image, CMD is never executed thus you get the error you have reported.

In the other hand, RUN instruction executes during build time making its result available for the next step. Quoting again from [docs](#):

The RUN instruction will execute any commands in a new layer on top of the current image and commit the results. The resulting committed image will be used for the next step in the Dockerfile.

It should be clear by now why the multistage build completes successfully when RUN is used in contrast to CMD instruction.

Your confusion started from wrongly assuming that the below is true:

I thought in the intermittent container, the cmd would execute which should make both the commands equivalent right?

Highest Rated Answer: None