

Title: Problem with using jib maven plugin offline for building docker container

Post Body:

I'm following Spring Boot Docker Tutorial from [here](#).

I got the exact result while having an internet connection. Now I need to produce the same result in an environment without an internet connection. I copied maven repositories and docker image into the new environment. I'm quite sure maven and docker is up and running.

When I tried to run following command `com.google.cloud.tools:jib-maven-plugin:dockerBuild -Dimage=eureka` I'm getting error messages. I guess there are some files plugin cannot locate but not sure which ones.

I'm adding the error message

```
[INFO] Scanning for projects... [INFO] [INFO] -----< com.ays:eureka >----- [INFO]
```

Can somebody point me where should I look or is there anything I'm missing at this point?

Here is my DockerFile

```
FROM openjdk:8-jdk-alpine RUN addgroup -S spring && adduser -S spring -G spring USER spring:spring ARG DEPENDENCY=target/dep
```

I didn't change anything in DockerFile.

Accepted Answer:

If you do not specify a base image, [Jib 3.0+ by default uses adoptopenjdk:8-jre](#) as a base image for Java 8 apps. (Prior to 3.0, Jib used `gcr.io/distroless/java:8` as the default.)

When you are not using a specific image digest (such as `gcr.io/distroless/java@sha256:...`) but instead use a tag (`:8` in your case with Jib < 3.0) for a base image, the tag can point to different images over time. That is, if you build an image sometime later on a different machine, Jib may pick up a slightly different base image than you used before. Hence the following warning:

```
[WARNING] Base image 'gcr.io/distroless/java:8' does not use a specific image digest - build may not be reproducible
```

For this reason, when you are not using a digest, Jib reaches out to the registry (`gcr.io`) and checks if the locally cached image (not in the local Docker engine cache but Jib's own cache) is up to date. If not, Jib will download the updated image. This is why you are getting the error when you're offline.

You have two options.

Pass `--offline` to Maven on the command-line. Then, Jib will use the cached base image; there will be no online connection. Of course, for this to work, Jib should have cached the base image before; you need to run Jib at least once online beforehand.

Use a digest to pin down a specific base image. For example, in your `pom.xml`,

```
<configuration>    <from><image>adoptopenjdk@sha256:9cb8bc7356ec2d9de56f3be3d8204a846ca0a3220af059aa67c35d53c7138e17</image>
```

If you prefer, you can specify both tag and digest. However, the tag will have no effect and only serve as a comment in this case.

```
<from><image>adoptopenjdk:8-jre@sha256:9cb8bc7356ec2d9de56f3be3d8204a846ca0a3220af059aa67c35d53c7138e17</image></from>
```

To find out which digest to use, check the [adoptopenjdk Docker Hub repository](#). Another way is to run Jib once online. After the warning, you will see a message reporting the current digest for the tag.

```
[WARNING] Base image 'adoptopenjdk:8-jre' does not use a specific image digest - build may not be reproducible [INFO] Using
```

Another option when you're running a local Docker daemon is to make Jib use an image from the daemon by prefixing `docker://` to the base image (for example, `<image>docker://openjdk:11-jre-slim</image>`). However, depending on circumstances, this can be a little bit slower than using a remote base image (but probably not so much even in that case).

Lastly, ***you can delete your Dockerfile***. Jib does not use Dockerfile, Docker CLI, or Docker daemon.

Highest Rated Answer: None