Title: How can I use Jar produced by mvn package in another Jenkins declarative pipeline stage?

Post Body:

I'm trying to build CI/CD pipeline with Jenkins for Maven project. I can't seem to find any decent examples on how to use .jar file produced by mvn package in another Jenkins declarative pipeline stage. I need the jar file to make an docker-image before uploading it to docker-registry. Here's my relevant parts of jenkinsfile:

```
pipeline {   agent none   stages{     stage('Build Jar'){         agent {           docker {             image 'maven:3-alpine
```

Dockerfile:

```
#install OS FROM centos #install java RUN yum install -y java #make directory structure to store temporary files RUN mkdir -p
```

EDIT 1---:

Laszlos answer did the trick, as I noticed that my .jar file was under different name than dockerfile assumed. Here is my working final jenkinsfile:

```
pipeline {   agent none   stages{     stage('Build Jar'){         agent {           docker {             image 'maven:3-alpine
```

And modified dockerfile:

```
#install OS FROM centos #install java RUN yum install -y java #make directory structure to store temporary files RUN mkdir -p
```

Accepted Answer:

You are stashing the jar files under the name 'targetfiles'. This is good.

What I think is missing is to pop from the stash, try adding the unstash line as shown bellow. The files will be in the exact same location as when you stashed them. Under 'target/' in your case.

```
stage('Deploy') {   steps {         script{             unstash 'targetfiles'             sh 'docker build image-name:test'
```

Highest Rated Answer:

You can use the same dockerfile for all stages and just install docker cmd on it and mount the docker.sock to it.

Add this to the Dockerfile:

```
RUN apt-get update && \     apt-get install -y apt-transport-https ca-certificates curl software-properties-common && \     cu
```

And run set the Jenkinsfile like this:

```
pipeline {     agent {        docker {          image 'maven:3-alpine'          args '-u root -v /root/.m2:/root/.m2 -v /var/r
```