

Title: How to pass private repository credentials to maven docker image when using Google Cloud Build

Post Body:

I am trying to use Google Cloud Build to build my Java app. It allows to use so called cloud builders - docker images of different builders. I am using Maven. So the problem is that I have to use a private repository (artifactory) to deploy artifacts. This rep is password protected and I do not know how to pass these credentials to GC maven docker container.

I see that the only possible way is:

To run the shell script which will update the maven container settings.xml with something like:

```
<servers>      <server>      <id>myRepoName</id>      <username>${server.username}</username>      <password>${serv
```

set env variables in the cloudbuild.yml

Are there any other elegant ways to achieve what I'm trying to?

Accepted Answer:

I solved this by doing the following:

Create a Google Cloud Storage bucket and upload your desired settings.xml. I'm using GitHub Packages, following [their documentation](#)

Setup your cloudbuild.yaml with the following:

```
steps:  - name: gcr.io/cloud-builders/gsutil      args: ['cp', 'gs://ci-maven/settings.xml', 'settings.xml']  - name: maven:3
```

First, it copies the settings.xml to the current directory (/workspace). Then, using the Docker Maven image directly, we add --settings /workspace/settings.xml to our args to specify the settings.xml location. From there, Google Cloud Build was able to pull my private GitHub package to properly install my project.

It may be possible to copy to /usr/share/maven/ref/ in the first step to allow the default Maven Docker behavior, but I was not able to get this to work. If anyone does, let me know!

Based on [this answer to a slightly different question about caching artifacts](#) and [Google Cloud Build documentation](#)

Highest Rated Answer: None