Docker and private modules

To install private npm packages in a Docker container, you will need to use Docker's build-time variables.

Background: runtime variables

If you had the following Dockerfile:

```
FROM risingstack/alpine:3.3-v4.3.1-3.0.1

COPY package.json package.json
RUN npm install

# Add your source files
COPY . .
CMD npm start
```

Which will use the RisingStack Alpine Node.JS Docker image, copy the **`package.json`** into our container, installs dependencies, copies the source files and runs the start command as specified in the **`package.json`** .

In order to install private packages, you may think that we could just add a line before we run **`npm install`** , using the ENV parameter:

```
ENV NPM_TOKEN=00000000-0000-0000-0000-000000000000
```

However, this doesn't work as you would expect, because you want the npm install to occur when you run **`docker build`** , and in this instance, **`ENV`** variables aren't used, they are set for runtime only.

Using build-time variables in Docker

Instead of run-time variables, you must use a different way of passing environment variables to Docker, available since Docker 1.9: the ARG parameter.

Overview

1. Create and check in a project-specific .npmrc file
2. Update the Dockerfile
3. Build the Docker image

Create and check in a project-specific .npmrc file

A complete example that will allow you to use `--build-arg` to pass in your NPM_TOKEN requires adding a `.npmrc` file to the project.

Use a project-specific `.npmrc` file with a variable for your token to securely authenticate your Docker image with npm.

1. In the root directory of your project, create a custom `.npmrc` file with the following contents:

```
//registry.npmjs.org/:_authToken=${NPM_TOKEN}
```

1. Check in the `.npmrc` file.

Update the Dockerfile

The Dockerfile that takes advantage of this has a few more lines in it than the earlier example that allows us to use the `.npmrc` file and the `ARG` parameter:

```
FROM risingstack/alpine:3.3-v4.3.1-3.0.1

ARG NPM_TOKEN
COPY .npmrc .npmrc
COPY package.json package.json
RUN npm install
RUN rm -f .npmrc

# Add your source files
COPY . .
CMD npm start
```

This adds the expected `ARG NPM_TOKEN`, but also copies the `.npmrc` file, and removes it when `npm install` completes.

Build the Docker image

To build the image using the above Dockerfile and the npm authentication token, you can run the following command. Note the `.` at the end to give `docker build` the current directory as an argument.

```
docker build --build-arg NPM_TOKEN=${NPM_TOKEN} .
```

This will build the Docker image with the current **`NPM_TOKEN`** environment variable, so you can run **`npm install`** inside your container as the current logged-in user.

> **Note:** Even if you delete the **`.npmrc`** file, it will be kept in the commit history. To clean your secrets entirely, make sure to squash them.

< Using private packages in a CI/CD workflow