Title: Publish Docker images using Spring Boot Plugin without credentials

Post Body:

I've got a Gradle project with Spring Boot plugin and am trying to publish the image built by the plugin: `gradle bootBuildImage --publishImage`

The problem is to publish "either token or username/password must be provided" and we can't do that since we have different authentication mechanism in different environments. For example on local machine we're using `ecr-credentials-helper` and the pipeline uses `aws ecr get-login-token | docker login`.

Is there any way to force the plugin to let the docker handles the authentication? (I'm assuming the plugin uses the docker daemon on the host).

Currently I wrote a task to generate a token file using `aws ecr get-login-token` and read the token file in `bootBuildImage` task. But I don't like this solution, due to security reasons.

Accepted Answer: None

Highest Rated Answer:

Here's a solution that replicates `aws ecr get-login-password` within Gradle, using the AWS Java SDK. Although you could instead invoke the CLI directly from Gradle, this makes the build script more fragile, as it then depends on having a certain version of the CLI installed. Particularly so since ecr login was a breaking change between v1 and v2 of the CLI.

This assumes you have your AWS credentials set up in some standard way that the default credentials provider in the SDK will locate.

```
import software.amazon.awssdk.services.ecr.EcrClient  buildscript {     repositories {         mavenCentral()     }     depend
```