Updating Docker Images

How to update Banner Applications from updated Docker images

Marianne Gillfillan, Senior DBA & Cloud Architect

October 2022

Strata Information Group
3935 Harney St., Suite 203
San Diego, CA 92110

SIG

Contents

1. Updating Images

There comes a time in the life of every application when certain components need to be upgraded. Some of these upgrades can take a significant amount of labor and time to perform the same tasks over and over again across numerous hosts.

Containerized applications provide the environment whereby this type of upgrade cycle can be done in just a few short steps and then put into effect by just restarting the application. This will save time and labor and open up resources to do many other tasks.

In this exercise, we'll explore the steps needed to upgrade a Banner application that needs both a Tomcat upgrade and a JDK version upgrade. Manually, these tasks could take weeks to perform across the entire Banner framework. However, we'll show this can be done with just a few updates to a couple lines of code.

Let's get started.

1.1. Updating the Tomcat Image

It's time to upgrade Tomcat from one version to another. Seems like that might be a daunting task. Not really.

For this exercise, we're going to upgrade the Tomcat image from using JDK 8 to JDK 11, something that will be required for the Grails 5 upgrade. Just for kicks, we may even upgrade from Tomcat 8 to Tomcat 9.

Open the Tomcat Dockerfile and find the line where the version of Tomcat is specified.

tomcat1/Dockerfile

```
FROM tomcat:8.5-jdk8-openjdk
```

Go out to Docker Hub and search for the official tomcat image page

https://hub.docker.com/_/tomcat

Search for the latest version of Tomcat, for the JDK 11 openjdk version

- Click on the Tags tab
- In the Filter box type: 9-jdk11-openjdk
- Scroll down until you find the right image

- Click on the identified image link
- Verify the JDK and tomcat versions

```
ENV JAVA_VERSION=11.0.16

ENV TOMCAT_VERSION=9.0.65
```

Looks like we found what we need. Make note of the image name as that is what we'll need to update our tomcat image.

```
tomcat:9-jdk11-openjdk
```

Now that know what official Docker image to reference, replace the value in the tomcat Dockerfile with what we just found:

tomcat1/Dockerfile

```
FROM tomcat:9-jdk11-openjdk
```

Now rebuild your image with a new tag showing the version of Tomcat if you want. Then verify that your image was created.

```
. buildspec.sh mytomcat 9.0.65 1

docker images
...
mytomcat              9.0.65-1   1370ebfdaaf4   8 minutes ago   824MB
```

Terrific! We have an new Tomcat image with an updated version of Tomcat AND an updated version of JDK that we can use with Banner applications. WOW - that was easy!

Let's move on and upgrade our Banner container.

1.2. Updating the Banner Containers

Open the Dockerfile for the Banner application and find the FROM line where the tomcat image is specified.

applicatonNavigator/Dockerfile

```
FROM mytomcat:8.5.82-2
```

Change the version to the new image version just created:

applicatonNavigator/Dockerfile

```
FROM mytomcat:9.0.65-1
```

Save the file and rebuild the container.

```
. build.sh applicationNavigator appnav 3.7 8081
```

Check the log file and see how the container build went

```
docker logs applicationNavigator -f

** warning - no files are being watched **
bin/catalina.sh: line 421: /usr/local/openjdk-8/bin/java: No such file or
 ↪  directory
```

Something went wrong. Apparently, we missed changing an environment variable somewhere. Let's take care of that.

tomcat1/Dockerfile

```
ENV TOMCAT_JAVA_HOME="/usr/local/openjdk-11" \
```

Let's try that build again and check the logs

This time it turned out better. Let's jump in the container and check a few things.

- Check the environment variables
- Check the /usr/local/openjdk-11 to make sure it exists
- Run bin/version.sh to check the version of tomcat
- The jar files will work fine with JDK 11 so no need to update those.

If everything looks good - congrats - you just upgraded Tomcat and JDK! Rinse/Repeat for every other application.

1.3. Application Update recap

- Update FROM version
- Update JAVA_HOME environment variable
- Rebuild container

That's all there is to it. Now you can plan your next vacation with all the extra time you'll be saving running your applications in containers.

There are many more topics related to running applications in containers. These lessons just touch on a few of the basic topics to get with which to get started.

Some advanced topics include:

- Container Hardening
- Docker Swarm
- Docker Compose

There are a lot of Banner schools at this point running their Banner applications in containers. If you ever get stuck, post a message to the eCommunities or the BannerDBA list. Whether you choose to run containers on-prem or in the cloud, someone has already done it and has probably already run into the problem you're probably having. So never be afraid to ask.

Containers are also a great way to test out new things, just because they are so disposable. Create, destroy, modify, repeat, until you are completely satisified.