QA Consulting

Docker Shell Vs Exec

TRAINING MATERIALS - MODULE HANDOUT

Contacts

robert.crutchley@qa.com
team.qac.all.trainers@qa.com

www.consulting.qa.com

Contents

Overview

Some of the instructions, such as RUN, CMD and ENTRYPOINT allow you to either use the shell or exec forms for running commands when building a Docker image. The preferred method is to use the exec way, so that an unnecessary sub shell isn't made, this is explained below.

Basically going forward, use **exec** form:

```
CMD ["/usr/bin/java", "-jar", "/app.jar"]
```

Instead of the **shell** form:

```
CMD java -jar app.jar
```

The Details & Reasoning

When using the **shell** form, Docker will run this as a shell command when the container starts up, like this:

```
/bin/sh -c "java -jar app.jar"
```

The first process of the container when it starts running will be a shell **/bin/sh**, which will then start the java process. The shell in this is unnecessary, when you run the **docker ps** command, you will be able to see under the COMMAND column how the initial process has been started.

Environment Variables Will Not Work with []

Because you are not executing a shell, but a process directly, you will not be able to use environment variables with double braces ([]). For example, if you had a JAR file that you wanted to run in the home folder you could run the example below with the shell format, this would not work with brackets though:

```
CMD java -jar $HOME/app.jar
```

Tasks

This exercise will allow you to observe how processes are executed differently using the exec form vs the shell form.

- Create a new folder for the exercise, such as **~/docker/shell_vs_exec_exercise**.
- In that folder create a Dockerfile containing the following:

```
FROM alpine:latest
CMD ping google.com
```

- Build an image from this Dockerfile and run it in a container.
- Run **docker ps** and you will be able to see under the **COMMAND** column for you container that the initial process was started as a shell.

```
/bin/sh -c ‘ping google.com’
```

- Edit the Dockerfile so that it looks like below. Note that we have to give the full path to the application that we want to run because we are no longer using the shell, this means that we cannot utilise the PATH environment variable.

```
FROM alpine:latest
CMD ["/bin/ping", "google.com"]
```

- Build a new Docker image and run it in a container.
- Run **docker ps** again and you should be able to see that shell was not used this time.

```
/bin/ping google.com
```

- Stop and delete all the containers that you created
- Remove all of the images that you created