**TRAINING MATERIALS - MODULE HANDOUT**

**Contacts**

**robert.crutchley@qa.com**

*team.qac.all.trainers@qa.com*

*www.consulting.qa.com*

**Contents**

[**Overview**](#_3l2qekkrkn3k) **1**

[**Multistage Build with Java & Maven**](#_hb1fwnubisz2) **1**

[How Java and Maven work](#_tl0fluurjbej) 1

[Our options](#_a95ejzddhp5c) 1

[**Tasks**](#_lxj4n977dbal) **2**

[Overview](#_rfykmug4wh68) 2

[Creating the Java Application](#_5qao38xobmsv) 2

[Create the Static Webpage](#_xm01pbigy14) 3

[Create the Config File for Maven](#_8oo4smer8713) 3

[Create the Dockerfile](#_7by9n5a6qg5d) 4

[Clean Up](#_3m65ubpiru9m) 5

# Overview

Likely the reason your learning about Docker is to be able to deploy an application into it and get all the benefits from Docker. There are a some good practices adhere to when developing applications with Docker, multi stage builds is one of them. Multistage builds are aimed at keeping the image size for your application down. Another thing we want to be looking for when building and running applications is consistency between environments such as a developer environment and a build server environment, Multistage Builds will also aid us with this.

# Multistage Build with Java & Maven

## How Java and Maven work

We’ll be using the Java programming language and the Maven dependency and build tool for usage examples. When we want to build a Java application it will need to be compiled and packed into a JAR file so that it can be executed by Java, Maven does this for us.

## Our options

This means we have to decide how we are going to get a JAR file running in a Docker container, there are 3 options below and we’ll be going for number 3.

1. Build the project on the host machine with Maven, use the Dockerfile to copy the built JAR file into the Docker Image, set an entrypoint that runs the JAR file.  
   This requires Maven and Java to be installed on the host machine. Host machine environments can vary and effect the build.
2. In the Dockerfile copy everything into the image, build the project and then run the JAR file in an entrypoint.  
   Unless we delete it, all the dependencies for build the project will be in the .m2 folder for Maven and the source code will be sat in the image for no good reason as well.
3. Use a Multistage build. Build from a Maven image, copy the code and build it, Build from a Java image, copy the JAR file from the previous build stage, create an entrypoint to run the JAR file.  
   This means that Docker will build the Image the same everywhere, whether its a developers laptop or a build server. Also the image which runs the application will be kept as small as possible.