

**Using Docker for JVM (Java Virtual Machine) Projects**

**Course Number:** DVOP-156  
**Duration:** 0.5 days

**Overview**

This live, online Java Virtual Machine (JVM) projects training teaches attendees how to use Docker for their Java Virtual Machine (JVM) Projects to deliver reliable, containerized software as part of the automation process. After attending this course, students are prepared to immediately use Docker in their own JVM projects.

**Prerequisites**

* Experience with Java application development or any other JVM language (such as Groovy or Scala)
* A working knowledge of Docker, specifically using the Docker CLI to manage images (pull, push, create) and containers (create, run, stop), Dockerfiles, and Docker Compose
* Basic familiarity with JUnit 5 (useful but not required)

**Materials**

All attendees receive a copy of the instructor’s presentation and related code.

**Software Needed on Each Student PC**

* A machine (Linux, macOS, or Windows) with the latest Docker Community Edition, JDK 8 or higher, and IntelliJ Community Edition or Ultimate Edition installed
* Pull the Docker image openjdk:jre-alpine with the command docker pull openjdk:jre-alpine
* A Docker Hub account

**Objectives**

* Understand the tooling that helps with managing and abstracting Docker operations for JVM projects
* Implement typical developer workflows using Docker
* Build images for your JVM applications and distribute them to a Docker registry
* Write and execute different types of tests that use Docker containers as fixtures
* Automate the use of Docker from the IDE, the build process, and a CI/CD pipeline

**Outline**

* Introduction
  + What’s your experience with Docker as a Java developer?
  + A brief review of Docker concepts and terminology
  + Typical Docker workflows for JVM developers
* Containerization and Distribution of JVM Projects
  + Writing a Dockerfile for a Java application
  + Building an image for a Java application
  + Running the Java application in a container
  + Pushing an image to a registry;
  + Google Jib for abstracting the workflow
  + Docker CLI to produce an image
  + Running a previously built image in a container
  + Using the Docker CLI to push an image to Docker Hub
  + Using Jib to implement the containerization workflow
* Resilient, Reproducible Testing with Docker Containers
  + Different types of testing
  + Benefits of using Docker containers for testing
  + Introduction of TestContainers
  + Testing a multi-container application stack
  + Setting up, configuring, and using TestContainers with JUnit 5
  + Use TestContainers to write a test for a web service
  + Use Docker Compose with TestContainers
* Integrating Docker with JVM Tooling
  + Using Docker from the IDE (IntelliJ and Eclipse)
  + Using Docker from the build tool (Maven and Gradle)
  + Powering a Jenkins CI/CD pipeline with Docker
  + Run a container from IntelliJ Community Edition
  + Run a container from Maven or Gradle
  + Model Jenkins build pipeline steps that use Docker
* Conclusion