A Complete Guide to Docker for Operations and Development

Test-Prep for the Docker Certified Associate (DCA) Exam

—

Engy Fouda

Apress®

A Complete Guide to Docker for Operations and Development

Test-Prep for the Docker Certified Associate (DCA) Exam

Engy Fouda

Apress®

A Complete Guide to Docker for Operations and Development: Test-Prep for the Docker Certified Associate (DCA) Exam

Engy Fouda
Hopewell Junction, NY, USA

ISBN-13 (pbk): 978-1-4842-8116-1 ISBN-13 (electronic): 978-1-4842-8117-8
https://doi.org/10.1007/978-1-4842-8117-8

Copyright © 2022 by Engy Fouda

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Managing Director, Apress Media LLC: Welmoed Spahr
Acquisitions Editor: Aditee Mirashi
Development Editor: James Markham
Coordinating Editor: Aditee Mirashi

Cover designed by eStudioCalamar

Cover image designed by Freepik (www.freepik.com)

Distributed to the book trade worldwide by Springer Science+Business Media New York, 1 New York Plaza, Suite 4600, New York, NY 10004-1562, USA. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com. Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a **Delaware** corporation.

For information on translations, please e-mail booktranslations@springernature.com; for reprint, paperback, or audio rights, please e-mail bookpermissions@springernature.com.

Apress titles may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Print and eBook Bulk Sales web page at http://www.apress.com/bulk-sales.

Any source code or other supplementary material referenced by the author in this book is available to readers on GitHub via the book's product page, located at www.apress.com/978-1-4842-8116-1. For more detailed information, please visit http://www.apress.com/source-code.

Printed on acid-free paper

To my motivator and daughter, Areej, who talks about me
with proudness and makes me extremely happy.
To my husband, Hesham, who does everything he can do to support
me. To my hero and dad, Dr. Mohamed Samir, you are my role model.
Every single day, I try to be you. To my mom, Dr. Suzan, who knows
how to celebrate my achievements. To my brother, aunts, nieces,
nephews, and cousins, I wish I could celebrate this new book with you.

Table of Contents

About the Author xi

About the Technical Reviewer xiii

Part I: Docker Fundamentals 1

Chapter 1: Introduction.. 3

Who Should Read This Book? 3

How This Book Is Organized.. 3

Exam Topics 4

What You Will Learn 4

Summary.. 5

Chapter 2: Installation and Configuration.. 7

How to Install the Docker CE for Various Operating Systems 8

Difference Between Containerization and Virtualization.................................... 9

How to Set Up a Local Registry... 9

Configuration of Logging Drivers 10

The Usage of Namespaces and Cgroups 11

How to Set Up a Swarm, Configure Managers, Add Nodes, and Scale Your System.................. 11

First, Build a Simple Swarm of One Manager and One Worker 11

Second, Let Us Scale Up to Five Replicas.. 12

Summary... 13

Chapter 3: Image Creation, Management, and Registry 15

Docker Flow 15

Pulling an Image 16

Crafting a Container 18

Container Lifecycle 21

Create Your First Image 23

Push Your Images 27

Dockerfile Commands 29

The FROM Instruction 30

The MAINTAINER Instruction 30

The COPY Instruction 30

The ADD Instruction 30

The ENV Instruction 31

The USER Instruction 31

The WORKDIR Instruction 31

The VOLUME Instruction 31

The EXPOSE Instruction 32

The RUN Instruction 32

The CMD Instruction 32

The ENTRYPOINT Instruction 33

The inspect Command 33

Multistage Build 34

Summary 34

Chapter 4: Networking 35

Containers’ Isolated Networking 35

User-Defined Bridge Network 37

Exposing Container Ports 41

Some Differences in Networking Between Windows and Linux 44

Swarm Networking 46

What Is the Best Number of Managers? 50

Datacenter Topology 51

Replicated vs. Global Services 51

Port Publishing Modes 51

Kubernetes Networking 52

Summary 53

Chapter 5: Storage and Volumes 55

Persistent Volumes 55

Ephemeral Volumes 58

Data-Only Volume 60

Filesystem vs. Volume 62

Storage in Kubernetes 63

Summary 65

Chapter 6: Docker Enterprise Edition/Mirantis Kubernetes Engine Installation 67

Installing the Docker EE on Windows Server 2016 68

Testing the Installation 68

Installing the UCP 69

Step 1: Create the First Virtual Machine 70

Step 2: Initialize a Swarm and Add Nodes 72

Step 3: Install UCP Version 3.0.0 73

Step 4: Mount Issue 74

Step 5: Reinstall UCP Version 3.0.0 74

Step 6: Upload the License 76

Step 7: Install Kubernetes CLI 76

Step 8: Install the kubectl Binary via curl 78

Installing the DTR 80

Verify Logging from CLI from Both Nodes 82

Verify DTR Installation from UCP Web UI 84

Summary 85

Chapter 7: Universal Control Plane (UCP) 87
The Interface 88
Different UCP Versions 89
Switching Between Orchestrators 90
Docker Content Trust 92
Summary 93
Chapter 8: Docker Trusted Registry (DTR) 95
Install the DTR from the UCP 95
Enable/Disable Image Scanning 101
Push an Image to the DTR Repository 101
Immutable Images 102
Image Pruning 103
Summary 103
Chapter 9: Microservices 105
Microservices vs. Monolithic 105
Process of Containerizing an App 107
Python Example 108
Java Example 114
.NET Example 117
Extra Resources for .NET Tools 117
Ebook 118
Extra Examples 118
Summary 119
Chapter 10: Orchestration 121
From Development to Production 121
How to Write a Swarm docker-compose YAML File? 123
How to Write a Kubernetes Manifest YAML File? 127
Summary 134

Chapter 11: Security 135

Role-Based Access Control (RBAC) 135

Create a Subject 136

Create a Collection 142

Create a Role 150

Create a Grant 152

Docker Content Trust (DCT) 156

Transport Layer Security (TLS) 160

Summary 161

Chapter 12: DCA Exam Requirements 163

Examination Format 163

Pass the Exam Sections 163

Registration Link and Certification 164

For Online Remotely Proctored Exams 164

Question Types 165

Multiple-Choice Questions 165

Discrete Option Multiple-Choice (DOMC) Questions 165

Extra Resources 166

Part II: Exam Prep Quizzes 169

Chapter 13: Orchestration Quiz 171

Answer Key 174

Chapter 14: Image Creation, Management, and Registry Quiz 175

Answer Key 178

Chapter 15: Installation and Configuration Quiz 179

Answer Key 183

Chapter 16: Networking Quiz 185

Answer Key 189

Chapter 17: Security Quiz 191

Answer Key 192

Chapter 18: Storage and Volumes Quiz 193

Answer Key 196

Index 197

About the Author

Engy Fouda is an adjunct lecturer at SUNY New Paltz teaching Intro to Data Science using SAS Studio and Introduction to Machine Learning using Python. She is an Apress and Packt Publishing author. Currently, she teaches SAS, Docker, and Kubernetes tracks at ONLC (Microsoft Partner) and other venues as a freelance instructor. She also works as a freelance editor at the Polaris&Dawn company. She holds two master's degrees: one in journalism from Harvard University, the Extension School and the other in computer engineering from Cairo University. Moreover, she earned a Data Science Graduate Professional Certificate from Harvard University, the Extension School. She volunteers as the executive manager and is a former team leader for Momken Group (Engineering for the Blind), Egypt Scholars Inc.

About the Technical Reviewer

Onur Yilmaz is a senior software engineer at a multinational enterprise software company. He is a Certified Kubernetes Administrator (CKA) and works on Kubernetes and cloud management systems as a keen supporter of cutting-edge technologies. Furthermore, he is the author of multiple books on Kubernetes, Docker, serverless architectures, and cloud-native continuous integration and delivery. In addition, he has one master's and two bachelor's degrees in the engineering field.

PART I

Docker Fundamentals

CHAPTER 1

Introduction

The book's scope is to help you pass the Docker Certified Associate (DCA) exam. Earning the certificate is proof of your experience in working with containers; the Mirantis Kubernetes Engine (MKE), formerly known as the Docker Enterprise Engine; swarms; clusters; and containerized applications.

Getting certified sets your minimum salary and helps you present yourself in the market. Moreover, plenty of corporations set being certified as a condition to get hired.

This prep book will cover all the topics that you will be tested on during the exam and more. You will learn how to install and use the Universal Control Plane (UCP) and the Docker Trusted Registry (DTR). Moreover, it has plenty of real exam questions from previous exams to practice. This book will help you identify your strengths and weaknesses and guide you on how to increase your knowledge in the weak topics.

Who Should Read This Book?

The book's primary targets are system administrators, operations managers, developers, and IT professionals who would seek to pass the DCA exam. The book assumes that you have some experience with Docker and Kubernetes and are looking to get certified.

However, the book is helpful for anyone looking for these topics in-depth and learning them. It provides detailed explanations of the concepts with hands-on examples in a step-by-step format.

How This Book Is Organized

The book is divided into two parts and follows the main outline of the exam sections. However, it covers even more than that. In Part 1, Chapters 1–12, you will find some extra chapters covering topics you will need in your practical workdays on which you will not be tested. Those chapters providing extra information will not have quizzes for them.

© Engy Fouda 2022

E. Fouda, *A Complete Guide to Docker for Operations and Development*,
https://doi.org/10.1007/978-1-4842-8117-8_1

In Part 2, Chapters 13–18, you will find quizzes for the chapters that are included in the exam only.

Chapters 2–5, 10, and 11 all cover exam sections. Chapters 6–9 are not explicitly in the exam, but you cannot work without knowing this information. Chapter 12 tries to focus on the exam steps and how to register. Chapters 13–18 have a sample of real exam questions.

Exam Topics

The exam covers the following topics:

- Orchestration (25% of the exam)
- Image creation, management, and registry (20% of the exam)
- Installation and configuration (15% of the exam)
- Networking (15% of the exam)
- Security (15% of the exam)
- Storage and volumes (10% of the exam)

You must pass every section of the exam and get more than 50% in the total. If you failed any section, you would fail the whole test.

What You Will Learn

- Learn the difference between the container and the virtual machine, the lifecycle of the container, and the various stages of development.
- Know how to install Docker on various platforms and how to manage the resources.
- Create your first container and image. Push and pull to and from the registry hub. Learn how to install your own local registry.
- Mount volumes.

- Learn all about container networking.
- Orchestrate between containers and learn how to debug your containers.
- Learn how to create a swarm and prepare your containers for production.
- Write a Dockerfile, Docker Compose YAML file, and Kubernetes manifest YAML file.
- Learn how to install the Docker Enterprise Edition (EE)/Mirantis Kubernetes Engine (MKE) with client shell.
- Learn how Docker achieves agility, portability, and control for developers and the IT operations team across all stages of the app lifecycle.
- Learn all about the exam and how to pass it, and work on plenty of real exam questions from previous exams to practice.

Summary

This chapter provided an overview of the road map to getting certified. It listed how the book is organized, the target audience, and the exam sections and their coverage percentages.

CHAPTER 2

Installation and Configuration

There are two Docker editions: Community Edition (CE, entirely free) and Enterprise Edition (EE, not free). This chapter will focus on Docker Community Edition (CE) installation. Chapters 6, 7, and 8 will show in detail how to install, configure, and use the Docker Enterprise Edition and its components. This chapter lists the common configuration commands between the two editions. Many of the exam questions test you on these common commands.

It is crucial to mention that the installation and configuration exam section is covered in this chapter and the chapters mentioned earlier. Moreover, this chapter's sections are not interconnected, but all lie under the installation and configuration exam section.

The chapter will cover the following topics:

1. How to install the Docker CE for various operating systems
2. The difference between containerization and virtualization
3. How to set up a local registry
4. Configuration of logging drivers
5. How to set up a swarm, configure managers, add nodes, and scale your system
6. The usage of namespaces and cgroups

© Engy Fouda 2022

E. Fouda, *A Complete Guide to Docker for Operations and Development*,
https://doi.org/10.1007/978-1-4842-8117-8_2

How to Install the Docker CE for Various Operating Systems

Docker has editions that can be installed on various operating systems. They are available on Docker Hub: `https://hub.docker.com/search?type=edition&offering=community`.

For Ubuntu, use this command: `wget -qO- https://get.docker.com/ | sh`.

For Microsoft Windows, enable Hyper-V and enable virtualization. If virtualization is not enabled, enter the computer BIOS, and enable it. For Windows 10 Professional or Enterprise 64-bit or Windows 10 Home 64-bit with WSL 2, install Docker Desktop for Windows. It is a pretty straightforward step: download the exe file and follow the instructions.

However, for Windows 7, 8, and 10 Home, use Docker Toolbox, and use Oracle VirtualBox. Every time, you'll need to click the Kitematic (Alpha) icon, as in Figure 2-1.

Figure 2-1. *Kitematic (Alpha) icon*

After Kitematic starts successfully, double-click the Docker Quickstart Terminal icon, as in Figure 2-2.

Figure 2-2. *Docker Quickstart Terminal*

If the host's operating system is Windows and Ubuntu is installed inside a virtual machine on top of it, avoid the "VboxClient Fail To Start" error on Ubuntu VirtualBox guest machines by following this link:

https://websiteforstudents.com/how-to-fix-vboxclient-fail-to-start-error-on-ubuntu-virtalbox-guest-machines/

There are no questions in the DCA exam about this point, but it can cause issues in the installation, causing you to not be able to do the rest of the book's hands-on labs.

Difference Between Containerization and Virtualization

Virtual machines represent hardware-level virtualization where one operating system is downloaded on top of a host operating system. Therefore, they are heavy and fully isolated. There is no availability for SSH or TTY, and they are fully secured.

On the other hand, containers represent operating system virtualization. You can download as many different operating systems in containers as applications need. Containers are light and fast; however, they are not fully isolated but secure, as discussed in Chapter 11.

How to Set Up a Local Registry

A Docker registry is a location where Docker images can be pushed, pulled, and stored. Registered users have their own spaces in the registry. This user-named space is called repository. A repository is a namespace that is used for storing images. For example, assume a user has a public image called `firstapp` that is pushed to Docker Hub. To reach to this image, one must use the image repository name, which is `<username>/<image name>`. In this example, assume that I am the image owner. Then the image repository is `engyfouda/firstapp`.

Registries can be hosted by a third party as a public or private registry. Docker Hub is an example of it. Later in the book, we will learn about the Docker Trusted Registry and how to install it and use it with the Docker EE. Also, there are Google Container Registry and AWS Container Registry.

Let us see how to set up and use a local registry. Setting it up is easy and fast. It is only one command as we will see. To test it, we will pull any image from Docker Hub, then push it to our local registry, delete all the local images, and pull it from the local registry.

To set up a local registry without a graphical user interface (GUI), we will use a repository image that is on Docker Hub called Registry with tag 2 (registry:2). The installation command is `docker run -d -p 5000:5000 --restart=always --name registry registry:2`. Basically, we craft a container based on the `registry:2` image. The `-d` option is to run as a daemon in the background. The `-p` is to set the <host port>:<container port>; we will talk about ports in detail in Chapter 4. We set the `restart` policy to be `always` and set the container name as `registry`.

To test this registry, we pull any image from Docker Hub, for example, `docker pull busybox`. Then rename it to add the local repository name using `docker tag busybox localhost:5000/firstapp`. Now, push it to the local registry using `docker push localhost:5000/firstapp`. Delete all the local images using `docker image remove busybox` and `docker image remove localhost:5000/firstapp`. To make sure that we do not have trace of these images, list the images using `docker image ls`, and we will not find these images in the list anymore. Pull the image from our local registry: `docker pull localhost:5000/firstapp`.

To remove the local registry, stop it and remove it using `docker container stop registry && docker container rm -v registry`.

To add a GUI, follow the steps in this site: http://joxit.github.io/docker-registry-ui/.

Configuration of Logging Drivers

By default, Docker uses the json-file logging driver, which caches container logs as JSON internally. To catch the log, use the `docker logs <Container ID or name>` subcommand.

There are other drivers that Docker can use, for example, splunk, journald, and none to disable logging. To configure the logging driver, run the `docker run -it --log-driver <log driver> <image>` subcommand.

To fetch the driver type, run `docker inspect -f '{{.HostConfig.LogConfig.Type}}' <CONTAINER>`.