

The SQL Server DBA's Guide to Docker Containers

**Agile Deployment without
Infrastructure Lock-in**

Edwin M Sarmiento

Apress®

The SQL Server DBA's Guide to Docker Containers: Agile Deployment without Infrastructure Lock-in

Edwin M Sarmiento
Ottawa, ON, Canada

ISBN-13 (pbk): 978-1-4842-5825-5
<https://doi.org/10.1007/978-1-4842-5826-2>

ISBN-13 (electronic): 978-1-4842-5826-2

Copyright © 2020 by Edwin M Sarmiento

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: Jonathan Gennick
Development Editor: Laura Berendson
Coordinating Editor: Jill Balzano

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

Distributed to the book trade worldwide by Springer Science+Business Media New York, 233 Spring Street, 6th Floor, New York, NY 10013. 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 rights@apress.com, or visit <http://www.apress.com/rights-permissions>.

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/9781484258255. For more detailed information, please visit <http://www.apress.com/source-code>.

Printed on acid-free paper

This book is dedicated to all SQL Server DBAs and IT professionals who, for years, are stuck on the belief that they don't have what it takes to succeed – that you're not good enough, not smart enough, not technical enough, not talented enough.

It's about time to let go of those lies and start living the life you were meant to live.

This book was written by a person who was told he would not amount to anything.

Table of Contents

About the Author xiii

About the Technical Reviewer xv

Acknowledgments xvii

Introduction xix

Chapter 1: Introduction to Containers 1

 The Latest 30-Year-Old Technology..... 2

 Docker – The Company and the Container Runtime 3

 The Case for Containers..... 4

 Containers vs. Virtualization..... 4

 Microsoft, Containers, and SQL Server on Linux..... 7

 Summary..... 8

Chapter 2: Install and Configure Docker on Windows Server 9

 Minimum System Requirements..... 10

 Enable Containers Feature in Windows Server 2016 11

 Download and Install the Docker-Microsoft PackageManagement Provider 12

 Download and Install the Docker Package..... 13

 Verifying the Docker Engine Installation 14

 Docker Desktop for Windows..... 15

 Summary..... 18

Chapter 3: Install and Configure Docker on Linux 19

 Minimum System Requirements..... 20

 Basic Linux Commands You Need to Get Started..... 21

 Installing Docker on CentOS Linux..... 23

TABLE OF CONTENTS

Update the CentOS System Packages	23
Install the Required Dependencies	24
Add the Docker Stable Repository to Your System	25
Install Docker CE	27
Starting the Docker Daemon and Enabling on Startup	28
Installing Docker on Ubuntu Linux	29
Update the Ubuntu System Packages	29
Install the Required Dependencies for Connecting to a Repository over HTTPS	30
Add the Official GPG Key from Docker	31
Add the Docker Stable Repository to Your System	32
Install Docker CE	33
Verifying the Docker Engine Installation	34
Getting Rid of sudo?	36
Summary	38
Chapter 4: The Docker Ecosystem	39
Hello World: Docker Edition	40
How Docker Runs Containers	42
The Docker Ecosystem	44
Docker CLI	44
Docker Daemon	45
Docker Commands	45
Docker Images and Containers	46
Docker Hub	47
Exploring Docker Hub	49
Docker Image Naming Conventions	51
Running SQL Server on Windows Containers	56
Running SQL Server on Linux Containers	58
Beginning to Explore Containers and Images	59
Things to Consider	60
Connecting to the SQL Server Instance Remotely	63

Lifecycle of a Container	66
Summary.....	67
Chapter 5: Docker Images and Containers	69
A Quick Review (And Then Some...).....	70
Behind the Scenes of Pulling Images	73
Fat Manifest?.....	75
Image Manifest.....	79
Content Digest.....	81
Distribution Hash	82
Your Docker Host's Local Filesystem	85
Union Filesystem	85
Files, Directory Names, and Symbolic Links.....	85
Stitching Layers: Assembling the Image	90
How a Union FileSystem Works	91
Traversing the Layers	92
The Container Layer	97
Summary.....	98
Chapter 6: Managing and Administering Containers.....	99
Configuring the Docker Daemon	100
Working with dockerd.....	101
Working with daemon.json	101
Changing the Default Image and Container Directory	103
Stopping and Restarting the Docker Daemon	106
Configure the Docker Daemon for Remote Access on CentOS Linux	106
Configure Remote Access to the Docker Daemon with the <i>systemd</i> Unit File	107
Protect the Docker Daemon Socket by Configuring Mutual TLS Encryption.....	110
Update the daemon.json to Include the Settings for Mutual TLS Encryption	114
Enable Firewall Port to Allow Traffic to Docker Daemon.....	114
Testing Remote Client Connectivity	115
Cleanup	116

TABLE OF CONTENTS

Configure Docker CLI Bash Completion	117
Starting and Stopping Containers	118
Deleting Containers and Images	120
Interacting with Running Containers.....	121
Sharing Files Between Host and Container.....	122
Configuring Container Resources	124
Exploring the Container Logs	126
Reviewing the Status of the Docker Daemon	128
Summary.....	130
Chapter 7: Persisting Data.....	133
Stateful vs. Stateless Applications.....	134
Containers Are Ephemeral and Immutable.....	135
How Cloud Providers Implement Infrastructure- as-a-Service	136
Docker Volumes	139
Attach Volumes to Containers	143
To -v or to --mount	144
Sharing Volumes Between Containers?.....	145
Leveraging Volumes with SQL Server on Containers	146
Smart SQL Server In-Place Upgrade	152
Summary.....	156
Chapter 8: Working with SQL Server on Linux.....	157
SQL Server on Linux Architecture	158
Project Drawbridge.....	159
SQLOS and SQLPAL	160
Difference Between SQL Server on Windows vs. Linux	162
Installing SQL Server on Linux	163
Manual Installation	164
Unattended Installation.....	166
Configuring the Firewall.....	169

A Note on the Linux Firewalls and Docker	173
Configuring SQL Server on Linux	174
Enabling the SQL Server Agent.....	176
Configuring Default Database Data and Log Directory	177
Configuring Default Database Backup Directory	177
Enabling Trace Flags.....	178
Viewing All Instance-Level Configuration Settings.....	179
Working with the Filesystem.....	180
Important Directories in the Linux Filesystem	181
File Permissions.....	183
Assigning Permissions to Files and Directories	185
Writing a Simple Linux Bash Script.....	188
Running the Script.....	192
Passing Parameters to the Script.....	193
Adding Conditional Logic in Bash Scripts.....	194
Summary.....	198
Chapter 9: Creating Custom SQL Server on Windows Container Images.....	199
Creating Custom Docker Images.....	200
Exploring the <i>Dockerfile</i>	202
The File.....	203
The FROM Instruction	204
The LABEL Instruction	204
The RUN Instruction.....	205
The COPY Instruction	206
The ADD Instruction	208
The SHELL Instruction	209
The CMD Instruction	210
The ENTRYPOINT Instruction.....	211
CMD vs. RUN vs. SHELL vs. ENTRYPOINT.....	212

TABLE OF CONTENTS

The WORKDIR Instruction	213
The ENV Instruction.....	213
Using Escape Characters	214
Putting the Pieces Together with a <i>Dockerfile</i>	215
Build the Custom SQL Server on Windows Image.....	218
Optimizing the Dockerfile.....	222
Summary.....	227
Chapter 10: Creating Custom SQL Server on Linux Container Images.....	229
Additional <i>Dockerfile</i> Instructions	229
The EXPOSE Instruction.....	230
The VOLUME Instruction	232
The USER Instruction	232
Build the Custom SQL Server on Linux Image	233
Installing a Specific SQL Server Version.....	236
Creating a Custom Image That Runs As Non-root	238
Running a Container As Non-root	242
Non-root User and Docker Volumes	247
Optimizing the Dockerfile.....	250
Running Scripts Inside the Dockerfile.....	252
Install the SQL Server Command-Line Utilities.....	252
Write the Bash Script to Check for Backups.....	253
Run the Bash Script in the Last Instruction	257
Multi-stage Builds.....	259
Docker Compose and YAML Files.....	263
Installing Docker Compose	263
Creating a YAML File for Multi-container Apps	264
Summary.....	268

Chapter 11: Guide to Docker Networking for SQL Server DBAs	269
The docker0 Bridge.....	270
Virtual Network Adapters	273
Working with and Exposing Ports: Running Multiple SQL Server Instances on Linux via Containers.....	275
Using the Same Port on Multiple Containers	276
Intra-container Communications	280
Putting It All Together.....	283
Summary.....	285
Appendix A: Building a Personal Lab Environment.....	287
The Need for a Personal Lab.....	288
Software for the Lab Environment	290
Operating System	290
Virtualization Platform	291
Installing CentOS Linux Server.....	291
Installing Ubuntu Linux Server	302
Utilizing Public Cloud Resources.....	325
Provisioning a Linux Docker Host on Microsoft Azure	326
Provisioning a Linux Docker Host on AWS.....	326
Provisioning a Linux Docker Host on Google Cloud Platform	327
Linux Client Tools	328
Using SSH to Connect to Linux	328
Using WinSCP for Copying Files to Linux.....	330
SQL Server Client Tools	333
Azure Data Studio.....	334
mssql-cli.....	336
Summary.....	337
Index.....	339

About the Author



Edwin M Sarmiento is the Managing Director of 15C, a consulting company that specializes in designing, implementing, and managing high-availability infrastructures. Proud of his heritage as a Filipino, Edwin now lives in Ottawa, Canada, and specializes in high availability, disaster recovery, and system infrastructure. He is a former Microsoft Data Platform MVP of 12 years standing and is a Microsoft Certified Master for SQL Server. He is very passionate about technology and has interests in music, neuroscience, social psychology, professional and organizational development, leadership, and management matters when not working with databases. Edwin strives continually to live up to his primary mission statement: to help people and organizations grow and develop their full potential.

About the Technical Reviewer

Borbala Toth-Apathy is a database professional who's been fascinated with SQL for more than 15 years. She is currently a SQL Server database architect. She enjoys digging deep into anything and everything data related, especially about performance.

Acknowledgments

To my Creator, Jesus Christ. Your absolute belief in me compels me to believe in myself.

To my parents. You should have told me earlier that I won't be able to read my own handwriting. I should have invented a speech-to-text app years ago for people who don't have a sense of humor.

To my mom... who sent us to private school but couldn't take us to McDonald's. I wouldn't be able to tell the stories in Chapters 5 and 9 if you could.

To my dad... who didn't want me to become a pilot. I wouldn't be able to appreciate the complexities of flight travel and tell stories about them in Chapter 4 if I was flying the plane.

To Daniel, Earl, and Nerisse. I'm surrounded by your extraordinary creativity every single day. It's no wonder we end up making crazy stuff up like the "idea generator." I've learned so much about developing my own creativity just by hanging out in the *D.E.N.*

To the Worship Team at myChurch Canada... who urged me to use a digital audio workstation before I can play keys with the team. I really needed an introductory story for Chapter 6.

To my editors at Apress – Jonathan, Laura, and Jill. You make your authors look like rockstars (OK, just the writing part because I owe my parents for my good looks).

To my technical editor, Borbala. Your reviews and comments kept me tight on the technical details and helped make them clearer.

To Audi, Locky, Lexy, and Lodgi. You'll have others joining you soon.

Introduction

The year was 1999. My ex-girlfriend (now wife) and I were wrapping up the final touches in preparation for a new Internet cafe that we were about to open. I unboxed my brand-new US Robotics Sportster 56K dial-up modem, plugged it into my main computer, and waited to test my Internet connection. This was the moment that I had been waiting for.

During this time, I was still relatively new to the Internet. Having been introduced to it during my university days, I understood how powerful it was. My classmates and I used it to send physics lab reports to each other, either through email or simple FTP. I skipped libraries and started using Lycos and AltaVista to do my research. I used it to create a simple website on GeoCities that functioned like a resume. I knew I chanced upon a gold mine that no one in my university (nor my city) was fully aware of. But this was nothing quite like what I saw when I turned on my computer and connected to the Internet.

An ad on my Yahoo Mail popped up about a new service offering from Dialpad: free international phone calls worldwide from the comforts of your own desk. I initially thought this was insane. You see, back in those days, to make international phone calls, you would have to dial a number that connects you to a trunk line or an operator who will make the call on your behalf. Plus, it costed an average of \$5/minute to call a phone in San Francisco, United States, from Manila, Philippines. This Dialpad thing didn't make sense.

But I had got nothing to lose. I connected my headset to the computer, launched the Dialpad applet inside Netscape Navigator, and dialed my aunt's phone number in San Francisco. Within a few minutes, we were talking – I can hear her, she can hear me. I could not believe it. What I thought was an insane idea had become reality. And I couldn't wait to tell my parents. No more expensive phone calls to the United States.

But...

It's interesting how a technological innovation such as the Voice over Internet Protocol (VoIP) that started more than four decades ago (it was invented in 1973 but wasn't commercialized until the late 1990s) has created an entire industry. It has created job opportunities like VoIP engineers, network engineers, developers who write apps that leverage VoIP, and the like. It allowed applications that run on top of VoIP to go mainstream – the friend you called on Instagram, the conference call you hosted on Zoom or Skype, or that aunt you talked to on WhatsApp.

INTRODUCTION

Every technological innovation that solves a major business pain point goes through the same process of conception, adoption, and, finally, becoming common. In the case of VoIP, the expensive and cumbersome task of making an international phone call. We technology professionals should be on the constant lookout for the next new thing, applying critical thinking as to how it can create the next industry. We need to learn how to observe the winds, watch where it is going, and adjust our sails. Or else, we run the risk of becoming obsolete and not able to advance in our careers. Mainframes and FORTRAN, anyone?

But that doesn't mean you'll be recognized by your peers as the Oracle of Tech. In fact, quite the opposite. You'll be ridiculed, mocked, scorned, or even considered crazy. When I told my mom I could call the United States for free using the Dialpad service, she literally laughed. When I told my classmates in 1996 that robots and artificial intelligence will become mainstream, I was ridiculed. When I told my peers back in 2003 that they should look forward to having a Linux exam option added to the Microsoft Certified Systems Engineer certification, they called me crazy. It didn't help that Steve Ballmer called Linux a cancer in 2001.

This is why I described the lifecycle of a technological innovation. Because if you know how it works, it wouldn't matter what other people think or say. You are absolutely sure that it will work the way it does. Look back at the technological innovations in the last four decades. Virtualization was born out of a need to share expensive computer resources among a large group of users back in the days of the mainframe. Today, every organization leverages virtualization to run applications. The public cloud was born out of a need to treat IT as a business model to generate revenue instead of merely a cost of doing business. Today, public cloud providers are used instead of building private data centers and buying hardware.

This brings us to 2017, the year Microsoft released SQL Server 2017. It was the first version of SQL Server that ran on top of Linux. Yes, the same Linux that Steve Ballmer called a cancer in 2001 is now running the world's most popular operational database management system, according to Gartner. Even the Microsoft Azure certification exams have Linux components in them. Microsoft plus Linux is no longer a topic of April Fool's jokes. This is real. And just like the lifecycle of any technological innovation, this will become mainstream. The question is, "will you be ready when that time comes?"

Alongside the launch of SQL Server on Linux came another technological innovation already emerging: the container technology. While originally designed for the Linux operating system, containers solve a unique business problem that every IT organization

has: reliably run software when moved from one computing environment to another. Imagine upgrading or migrating from a physical machine to another physical machine or to a virtual machine or to a public cloud. Even moving code from a developer's laptop to a production environment. I'm sure you've heard your developers say "but it worked in my laptop" more than a dozen times. Containers are here to eliminate these challenges.

With both technologies already available, it's not surprising to see Microsoft not only support SQL Server on Linux running in container technologies but also leverage it in their own internal development and build processes. It's only a matter of time when SQL Server on Linux running on containers becomes more common than the SQL Server on Windows that we've known for the past two-and-a-half decades.

And that's the goal of this book. SQL Server database administrators and developers like yourself who have been working with SQL Server for a while already know a thing or two about the relational database engine. You already know how SQL Server stores data on data pages, how indexes speed up (or not) queries, and how it achieves high availability. I want you to take the same knowledge and experience as you prepare yourself for the next wave of technological innovation in container technologies. Imagine you're on a journey and this book will be your guide, each chapter laying out steps along the way toward your final destination.

How This Book Is Structured

This book consists of 11 chapters with an Appendix and is structured in the following way:

- **Chapter 1** introduces you to container technologies, a bit of a history of how it started and how Microsoft adopted it for the Windows Server operating system as well as SQL Server on Linux.
- **Chapter 2** walks you through how to configure a Windows Server operating system as a container host, allowing you to leverage your existing knowledge of Windows.
- **Chapter 3** covers a bit of what you need to know to get started with the Linux operating system in preparation for configuring it to run as a container host.
- **Chapter 4** covers the Docker ecosystem to better understand how container technologies work.

INTRODUCTION

- **Chapter 5** introduces you to Docker images and containers but with a focus on using them with SQL Server.
- **Chapter 6** is where managing and administering containers come in, introducing the most common operational tasks that you need to perform when working with containers.
- **Chapter 7** introduces the concept of persisting data with containers since SQL Server is designed to store and persist data.
- **Chapter 8** covers a bit of working with SQL Server on Linux, the installation and management experience to get you started with administering SQL Server on Linux and, eventually, building custom SQL Server on Linux container images.
- **Chapter 9** walks you through building a custom SQL Server on Windows container image that you can use for standardized deployments.
- **Chapter 10** walks you through building a custom SQL Server on Linux container image.
- **Chapter 11** covers a bit of Docker networking so you can better configure SQL Server on containers for interaction with other applications.

I highly recommend that you start with the **Appendix**. While it is very easy nowadays to spin up a lab environment on a public cloud infrastructure such as Amazon AWS, Microsoft Azure, or Google Cloud Platform, I don't want you to miss the experience of setting everything up from start to finish. I'm surprised to hear from senior SQL Server DBAs who have no experience installing a Windows Server operating system and configuring TCP/IP. There's nothing wrong with that. But, oftentimes, it's what determines whether an issue can easily be resolved or not.

Plus, because the network infrastructure in public cloud environments is offered as a service, there are several layers that abstract what's really going on. It can be very frustrating to try and test out something new in a safe environment only to find out that you wasted so many hours solving a problem that you have no control over or can only be solved by providing your credit card information. You're here to learn container technologies, not get into financial debt.

So, let's get started and dive into the world of container technologies.