

Vorwort

Container sind eine schlanke und portable Möglichkeit, beliebige Anwendungen und ihre Abhängigkeiten zu verpacken und transportabel zu machen.

Wenn man das so aufschreibt, klingt es ziemlich trocken und langweilig. Aber die Verbesserungen im Prozess, die durch Container möglich werden, sind das genaue Gegenteil davon – korrekt eingesetzt können Container wegweisend sein. Die Verlockungen der dadurch möglich werdenden Architekturen und Arbeitsabläufe sind so überzeugend, dass man glauben mag, spätestens in einem Jahr hat sich in jeder größeren IT-Firma der Status von »Docker? Nie gehört!« zu »Klar, schauen wir uns gerade an und setzen wir auch schon probetalb ein« geändert hat.

Der Aufstieg von Docker ist erstaunlich. Ich kann mich an keine Technologie erinnern, die so schnell so tiefgreifende Auswirkungen auf die IT-Branche hatte. Dieses Buch ist mein Versuch, Ihnen zu zeigen, *warum* Container so wichtig sind, *was* es Ihnen bringt, Ihre Umgebung zu containerisieren, und – am wichtigsten – *wie* Sie das erreichen.

Wer dieses Buch lesen sollte

Dieses Buch versucht, Docker ganzheitlich anzugehen, die Gründe für dessen Einsatz zu erklären und zu zeigen, wie man es nutzt und in einen Softwareentwicklungsworkflow integriert. Es behandelt den gesamten Software-Lifecycle – von der Entwicklung über die Produktion bis zur Wartung.

Ich habe versucht, möglichst wenige Annahmen über Sie als Leser zu treffen, und gehe nur davon aus, dass Sie grundlegende Kenntnisse von Linux und der Softwareentwicklung im Allgemeinen haben. Zielgruppe sind vor allem Softwareentwickler, Administratoren und Systemverwalter (insbesondere solche, die einen DevOps-Ansatz verfolgen wollen). Aber auch technisch affine Manager und andere Interessierte sollten ebenfalls Gewinn aus diesem Buch ziehen.

Warum ich dieses Buch geschrieben habe

Ich befand mich in der glücklichen Lage, Docker kennenzulernen und einzusetzen, während es noch ganz am Anfang seines kometenhaften Aufstiegs stand. Als sich die Gelegenheit ergab, dieses Buch zu schreiben, ergriff ich sie mit beiden Händen. Wenn mein Geschreibsel einem von Ihnen dabei hilft, die Containerisierungs-Community zu verstehen und das Beste daraus zu machen, dann habe ich mehr erreicht als in all den Jahren der Softwareentwicklung zuvor.

Ich hoffe wirklich, dass Ihnen das Lesen dieses Buches Spaß macht und es Ihnen dabei hilft, beim Einsatz von Docker in Ihrer Umgebung vorwärtszukommen.

Wie dieses Buch aufgebaut ist

Dieses Buch ist in drei Abschnitte unterteilt:

- Teil I beginnt mit der Erklärung, was Container sind und warum Sie an ihnen interessiert sein sollten. Dann folgt ein Tutorial-Kapitel mit den ersten Schritten beim Einsatz von Docker. Der Abschnitt endet mit einem langen Kapitel mit den zugrunde liegenden Konzepten und Technologien, die hinter Docker stecken, einschließlich eines Überblicks über die diversen Docker-Befehle.
- Teil II erklärt, wie Sie Docker in einem Softwareentwicklungs-Lifecycle einsetzen. Zuerst wird gezeigt, wie Sie eine Entwicklungsumgebung einrichten, bevor Sie eine einfache Webanwendung erstellen, die für den Rest dieses Abschnitts als Beispiel dient. Die Kapitel drehen sich um Entwicklung, Testen und Integration, aber auch darum, wie Sie Container ausrollen und ein Produktivsystem effektiv überwachen und protokollieren.
- In Teil III geht es um die fortgeschrittenen Details und die Tools und Techniken, die Sie brauchen, um Multihost-Cluster mit Docker-Containern sicher und zuverlässig zu betreiben. Haben Sie Docker schon im Einsatz und müssen herausfinden, wie Sie damit wachsen können oder Netzwerk- und Sicherheitsprobleme lösen, ist das Ihr Abschnitt.

Welchen Konventionen dieses Buch folgt

Die folgenden typografischen Konventionen werden in diesem Buch genutzt:

Kursiv

für neue Begriffe, URLs, E-Mail-Adressen, Dateinamen und Dateierweiterungen

Nichtproportionalschrift

für Programmlistings, aber auch für Codefragmente in Absätzen, wie zum Beispiel Variablen- oder Funktionsnamen, Datenbanken, Datentypen, Umgebungsvariablen, Anweisungen und Schlüsselwörter

fette Nichtproportionalschrift

für Befehle und anderen Text, der genau so vom Benutzer eingegeben werden sollte

kursive Nichtproportionalschrift

für Text, der vom Benutzer durch eigene Werte ersetzt werden sollte



Dieses Symbol steht für einen Tipp oder Vorschlag.



Dieses Symbol steht für einen allgemeinen Hinweis.



Dieses Symbol steht für eine Warnung oder Vorsichtsmaßnahme.

Codebeispiele

Zusätzliches Material (Codebeispiele, Übungen und so weiter) finden Sie (auf Englisch) zum Herunterladen auf <https://github.com/using-docker>.

Dieses Buch soll Ihnen bei der Arbeit helfen. Die Codebeispiele in diesem Buch können Sie im Allgemeinen in Ihren Programmen und Ihrer Dokumentation nutzen. Sie müssen uns nicht um Erlaubnis fragen, sofern Sie nicht einen signifikanten Anteil des Codes veröffentlichen. So erfordert zum Beispiel das Schreiben eines Programms, das diverse Codeschnipsel aus diesem Buch nutzt, keine Erlaubnis. Das Verkaufen oder Bereitstellen einer CD-ROM mit Beispielen aus diesem Buch benötigt hingegen eine Erlaubnis. Beantworten Sie eine Frage, indem Sie dieses Buch zitieren und Beispielcode daraus nutzen, benötigen Sie keine Erlaubnis. Übernehmen Sie einen deutlichen Teil des Beispielcodes für Ihre Produktdokumentation, müssen Sie dieses Vorhaben von uns genehmigen lassen.

Wir freuen uns über eine Quellenangabe, verlangen sie aber nicht unbedingt. Zu einer Quellenangabe gehören normalerweise Titel, Autor, Verlag und ISBN

(zum Beispiel: »Adrian Mouat: *Docker – Software entwickeln und deployen mit Containern*. dpunkt.verlag 2016, ISBN 978-3-86490-384-7«).

Wenn Sie das Gefühl haben, dass Ihr Einsatz der Codebeispiele über die Grenzen des Erlaubten hinausgeht, können Sie uns über hallo@dpunkt.de erreichen.

Danksagungen

Ich bin für all die Hilfe, Ratschläge und Kritik unglaublich dankbar, die ich während des Schreibens dieses Buches erhalten habe. Wenn ich einen Namen in der folgenden Liste vergessen habe, bitte ich um Verzeihung – ich wusste alle Beiträge zu schätzen, auch wenn ich nicht darauf reagiert habe.

Für allgemeines Feedback möchte ich mich bei Ally Hume, Tom Sugden, Lukasz Guminski, Tilaye Alemu, Sebastien Goasguen, Maxim Belouossov, Michael Boelen, Ksenia Burlachenko, Carlos Sanchez, Daniel Bryant, Christoffer Holmstedt, Mike Rathbun, Fabrizio Soppelsa, Yung-Jin Hu, Jouni Miikki und Dale Bewley bedanken.

Mein Dank für technische Diskussionen und Erläuterungen zu bestimmten Technologien geht an Andrew Kennedy, Peter White, Alex Pollitt, Fintan Ryan, Shaun Crampton, Spike Curtis, Alexis Richardson, Ilya Dmitrichenko, Casey Bisson, Thijs Schnitger, Sheng Liang, Timo Derstappen, Puja Abbassi, Alexander Larsson und Kelsey Hightower. Für die Erlaubnis, `monsterid.js` nutzen zu dürfen, bedanke ich mich bei Kevin Gaudin.

Ein großer Dank für all die Hilfe geht auch an die O'Reilly-Mitarbeiter, insbesondere an meinen Lektor Brian Anderson und an Meghan Blanchette, die das Ganze erst ins Rollen gebracht hat.

Diogo Mónica und Mark Coleman – vielen Dank an euch beide für das Beantworten meines Hilferufs in letzter Sekunde.

Zwei Firmen sollen besonders hervorgehoben werden: Container Solutions und CloudSoft. Jamie Dobson und Container Solutions sorgten dafür, dass ich mit Bloggen und Vorträgen beschäftigt war, und brachten mich mit vielen Leuten zusammen, die Einfluss auf dieses Buch hatten. CloudSoft erlaubten es mir freundlicherweise, in ihren Räumen an diesem Buch zu arbeiten. Zudem organisierten sie das Edinburgh Docker Meetup, das ebenfalls sehr wichtig für mich war.

Dafür, dass Ihr mein Gejammer über dieses Buch und meine Obsession ausgehalten habt, möchte ich mich bei all meinen Freunden und meiner Familie bedanken – Ihr wisst, wen ich meine (und werdet diese Zeilen sehr wahrscheinlich sowieso nicht lesen).

Schließlich möchte ich den DJs von BBC 6 Music danken, die den Soundtrack zu diesem Buch geliefert haben – unter anderem Lauren Laverne, Radcliffe and Maconie, Shaun Keaveny und Iggy Pop.