

marko LUKŠA



# kubernetes in action

Anwendungen in Kubernetes-Clustern  
bereitstellen und verwalten

HANSER



Im Internet: Github-Repository  
zum Buch



## Bleiben Sie auf dem Laufenden!



Unser **Computerbuch-Newsletter** informiert Sie monatlich über neue Bücher und Termine. Profitieren Sie auch von Gewinnspielen und exklusiven Leseproben. Gleich anmelden unter



[www.hanser-fachbuch.de/newsletter](http://www.hanser-fachbuch.de/newsletter)



**Hanser Update** ist der IT-Blog des Hanser Verlags mit Beiträgen und Praxistipps von unseren Autoren rund um die Themen Online Marketing, Webentwicklung, Programmierung, Softwareentwicklung sowie IT- und Projektmanagement. Lesen Sie mit und abonnieren Sie unsere News unter



[www.hanser-fachbuch.de/update](http://www.hanser-fachbuch.de/update)





Marko Lukša

# Kubernetes in Action

Anwendungen in Kubernetes-Clustern  
bereitstellen und verwalten

HANSER

Übersetzung: G & U Language & Publishing Services, Flensburg, [www.gundu.com](http://www.gundu.com)

Titel der Originalausgabe: „Kubernetes in Action“, © 2018 by Manning Publications Co.

Original English language edition published by Manning Publications USA © 2018 by Manning Publications.

German-language edition copyright © 2018 by Carl Hanser Verlag München. All rights reserved

Alle in diesem Buch enthaltenen Informationen, Verfahren und Darstellungen wurden nach bestem Wissen zusammengestellt und mit Sorgfalt getestet. Dennoch sind Fehler nicht ganz auszuschließen. Aus diesem Grund sind die im vorliegenden Buch enthaltenen Informationen mit keiner Verpflichtung oder Garantie irgendeiner Art verbunden. Autor und Verlag übernehmen infolgedessen keine juristische Verantwortung und werden keine daraus folgende oder sonstige Haftung übernehmen, die auf irgendeine Art aus der Benutzung dieser Informationen – oder Teilen davon – entsteht.

Ebenso übernehmen Autor und Verlag keine Gewähr dafür, dass beschriebene Verfahren usw. frei von Schutzrechten Dritter sind. Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Buch berechtigt deshalb auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.

Bibliografische Information der Deutschen Nationalbibliothek:

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

Dieses Werk ist urheberrechtlich geschützt.

Alle Rechte, auch die der Übersetzung, des Nachdruckes und der Vervielfältigung des Buches, oder Teilen daraus, vorbehalten. Kein Teil des Werkes darf ohne schriftliche Genehmigung des Verlages in irgendeiner Form (Fotokopie, Mikrofilm oder ein anderes Verfahren) – auch nicht für Zwecke der Unterrichtsgestaltung – reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

Copyright für die deutsche Ausgabe:

© 2018 Carl Hanser Verlag München, [www.hanser-fachbuch.de](http://www.hanser-fachbuch.de)

Lektorat: Sylvia Hasselbach

Copy editing: Sandra Gottmann, Münster-Nienberge

Umschlagdesign: Marc Müller-Bremer, München, [www.rebranding.de](http://www.rebranding.de)

Umschlagrealisation: Stephan Rönigk

Gesamtherstellung: Kösel, Krugzell

Ausstattung patentrechtlich geschützt. Kösel FD 351, Patent-Nr. 0748702

Printed in Germany

Print-ISBN: 978-3-446-45510-8

E-Book-ISBN: 978-3-446-45602-0

*Für meine Eltern, die die Bedürfnisse ihrer Kinder  
stets über ihre eigenen gestellt haben.*





# Inhalt

<b>Vorwort</b> .....	<b>XXI</b>
Der Autor .....	XXII
Danksagung zur englischsprachigen Ausgabe .....	XXIII
<b>Über dieses Buch</b> .....	<b>XXV</b>
Zielgruppe .....	XXV
Der Aufbau dieses Buches .....	XXVI
Der Code .....	XXVII
Das Forum zum Buch .....	XXVIII
Sonstige Onlinequellen .....	XXVIII
<b>Teil I: Überblick</b> .....	<b>1</b>
<b>1 Einführung in Kubernetes</b> .....	<b>3</b>
1.1 Der Bedarf für ein System wie Kubernetes .....	4
1.1.1 Von monolithischen Anwendungen zu Microservices .....	4
1.1.2 Eine konsistente Umgebung für Anwendungen bereitstellen ...	8
1.1.3 Übergang zu Continuous Delivery: DevOps und NoOps .....	8
1.2 Containertechnologien .....	10
1.2.1 Was sind Container? .....	10
1.2.2 Die Containerplattform Docker .....	14
1.2.3 Die Docker-Alternative rkt .....	18
1.3 Kubernetes .....	19
1.3.1 Die Ursprünge .....	19
1.3.2 Kubernetes im Überblick .....	19
1.3.3 Die Architektur eines Kubernetes-Clusters .....	21
1.3.4 Anwendungen auf Kubernetes ausführen .....	22
1.3.5 Vorteile der Verwendung von Kubernetes .....	24
1.4 Zusammenfassung .....	27

<b>2</b>	<b>Erste Schritte mit Docker und Kubernetes</b>	<b>29</b>
2.1	Containerimages mit Docker erstellen, ausführen und teilen	29
2.1.1	Docker installieren und einen Hello-world-Container ausführen	30
2.1.2	Eine triviale Node.js-Anwendung erstellen	32
2.1.3	Eine Docker-Datei für das Image erstellen	33
2.1.4	Das Containerimage erstellen	33
2.1.5	Das Containerimage ausführen	36
2.1.6	Das Innenleben eines laufenden Containers untersuchen	37
2.1.7	Container anhalten und entfernen	39
2.1.8	Das Image zu einer Registry hochladen	39
2.2	Kubernetes-Cluster einrichten	41
2.2.1	Einen lokalen Kubernetes-Cluster mit einem Knoten mithilfe von Minikube ausführen	41
2.2.2	Gehostete GKE-Cluster	43
2.2.3	Einen Alias und die Befehlszeilenvervollständigung für kubectl einrichten	46
2.3	Eine erste Anwendung in Kubernetes ausführen	47
2.3.1	Die Node.js-Anwendung bereitstellen	48
2.3.2	Auf die Webanwendung zugreifen	51
2.3.3	Die logischen Bestandteile des Systems	52
2.3.4	Anwendungen horizontal skalieren	54
2.3.5	Auf welchen Knoten läuft die Anwendung?	57
2.3.6	Das Kubernetes-Dashboard	58
2.4	Zusammenfassung	59
	<b>Teil II: Grundlagen</b>	<b>61</b>
<b>3</b>	<b>Pods: Container in Kubernetes ausführen</b>	<b>63</b>
3.1	Einführung in Pods	63
3.1.1	Wozu benötigen wir Pods?	64
3.1.2	Grundlagen von Pods	65
3.1.3	Container auf Pods verteilen	66
3.2	Pods aus YAML- und JSON-Deskriptoren erstellen	69
3.2.1	Den YAML-Deskriptor eines bestehenden Pods untersuchen	69
3.2.2	Einen einfachen YAML-Deskriptor für einen Pod schreiben	71
3.2.3	Einen Pod mit kubectl create erstellen	73
3.2.4	Anwendungsprotokolle anzeigen	74
3.2.5	Anforderungen an den Pod senden	75
3.3	Pods mithilfe von Labels ordnen	76
3.3.1	Einführung in Labels	77
3.3.2	Labels beim Erstellen eines Pods angeben	78
3.3.3	Labels vorhandener Pods ändern	79
3.4	Eine Auswahl der Pods mithilfe von Labelselektoren auflisten	80
3.4.1	Pods anhand eines Labelselektors auflisten	80

3.4.2	Labelselektoren mit mehreren Bedingungen .....	82
3.5	Die Podzuweisung mithilfe von Labels und Selektoren einschränken .....	82
3.5.1	Labels zur Klassifizierung von Arbeitsknoten .....	83
3.5.2	Pods bestimmten Knoten zuweisen .....	84
3.5.3	Zuweisung zu einem einzelnen Knoten .....	84
3.6	Pods mit Anmerkungen versehen .....	85
3.6.1	Die Anmerkungen zu einem Objekt einsehen .....	85
3.6.2	Anmerkungen hinzufügen und ändern .....	86
3.7	Ressourcen mithilfe von Namespaces gruppieren .....	86
3.7.1	Der Bedarf für Namespaces .....	87
3.7.2	Andere Namespaces und die zugehörigen Pods finden .....	87
3.7.3	Namespaces erstellen .....	88
3.7.4	Objekte in anderen Namespaces verwalten .....	89
3.7.5	Die Trennung der Namespaces .....	90
3.8	Pods stoppen und entfernen .....	90
3.8.1	Pods unter Angabe des Namens löschen .....	90
3.8.2	Pods mithilfe von Labelselektoren löschen .....	91
3.8.3	Pods durch Entfernen eines ganzen Namespaces löschen .....	91
3.8.4	Alle Pods in einem Namespace löschen und den Namespace erhalten .....	92
3.8.5	(Fast) alle Ressourcen in einem Namespace löschen .....	92
3.9	Zusammenfassung .....	93
<b>4</b>	<b>Replikationscontroller &amp; Co.: Verwaltete Pods bereitstellen .....</b>	<b>95</b>
4.1	Pods funktionsfähig halten .....	96
4.1.1	Aktivitätssonden .....	96
4.1.2	HTTP-Aktivitätssonden erstellen .....	97
4.1.3	Eine Aktivitätssonde in Aktion .....	98
4.1.4	Weitere Eigenschaften der Aktivitätssonde festlegen .....	99
4.1.5	Wirkungsvolle Aktivitätssonden erstellen .....	100
4.2	Replikationscontroller .....	102
4.2.1	Die Funktionsweise von Replikationscontrollern .....	103
4.2.2	Einen Replikationscontroller erstellen .....	105
4.2.3	Der Replikationscontroller in Aktion .....	106
4.2.4	Pods in den Gültigkeitsbereich eines Replikationscontrollers bringen und daraus entfernen .....	111
4.2.5	Das Pod-Template ändern .....	114
4.2.6	Pods horizontal skalieren .....	115
4.2.7	Einen Replikationscontroller löschen .....	117
4.3	Replikationssätze anstelle von Replikationscontrollern verwenden .....	118
4.3.1	Replikationssätze und Replikationscontroller im Vergleich .....	118
4.3.2	Einen Replikationssatz definieren .....	119
4.3.3	Einen Replikationssatz erstellen und untersuchen .....	120

4.3.4	Die ausdrucksstärkeren Labelselektoren des Replikationssatzes ..	121
4.3.5	Zusammenfassung: Replikationssätze .....	122
4.4	Daemonsets zur Ausführung einer Instanz eines Pods auf jedem Knoten ..	122
4.4.1	Einen Pod auf allen Knoten ausführen .....	122
4.4.2	Einen Pod nur auf einigen Knoten ausführen .....	123
4.5	Pods für endliche Aufgaben .....	126
4.5.1	Jobs .....	127
4.5.2	Einen Job definieren .....	128
4.5.3	Ein Job in Aktion .....	128
4.5.4	Mehrere Podinstanzen in einem Job ausführen .....	129
4.5.5	Die Zeit zum Abschließen eines Job-Pods begrenzen .....	130
4.6	Jobs regelmäßig oder zu einem späteren Zeitpunkt ausführen .....	131
4.6.1	Einen Cron-Job erstellen .....	131
4.6.2	Die Ausführung geplanter Jobs .....	132
4.7	Zusammenfassung .....	133
<b>5</b>	<b>Dienste: Pods finden und mit ihnen kommunizieren .....</b>	<b>135</b>
5.1	Dienste .....	136
5.1.1	Dienste erstellen .....	137
5.1.2	Dienste finden .....	143
5.2	Verbindungen zu Diensten außerhalb des Clusters .....	147
5.2.1	Dienstendpunkte .....	147
5.2.2	Manuell eingerichtete Dienstendpunkte .....	148
5.2.3	Einen Alias für einen externen Dienst erstellen .....	150
5.3	Dienste für externe Clients verfügbar machen .....	151
5.3.1	Einen NodePort-Dienst verwenden .....	151
5.3.2	Einen Dienst über einen externen Load Balancer verfügbar machen .....	155
5.3.3	Besondere Eigenschaften von externen Verbindungen .....	157
5.4	Dienste über eine Ingress-Ressource extern verfügbar machen .....	159
5.4.1	Eine Ingress-Ressource erstellen .....	161
5.4.2	Über den Ingress auf den Dienst zugreifen .....	162
5.4.3	Mehrere Dienste über denselben Domännennamen verfügbar machen .....	163
5.4.4	Einen Ingress für TLS-Datenverkehr einrichten .....	164
5.5	Die Bereitschaft eines Pods zur Annahme von Verbindungen signalisieren	166
5.5.1	Bereitschaftssonden .....	167
5.5.2	Einem Pod eine Bereitschaftssonde hinzufügen .....	168
5.5.3	Bereitschaftssonden in der Praxis .....	170
5.6	Headless-Dienste zur Ermittlung einzelner Pods .....	172
5.6.1	Einen headless-Dienst erstellen .....	172
5.6.2	Pods über DNS finden .....	173
5.6.3	Alle Pods finden – auch diejenigen, die nicht bereit sind .....	174

5.7	Fehlerbehebung bei Diensten .....	175
5.8	Zusammenfassung .....	176
<b>6</b>	<b>Volumes: Festplattenspeicher zu Containern hinzufügen .....</b>	<b>177</b>
6.1	Volumes .....	178
6.1.1	Ein Beispiel .....	178
6.1.2	Arten von Volumes .....	180
6.2	Gemeinsame Datennutzung durch die Container .....	181
6.2.1	emptyDir-Volumes .....	181
6.2.2	Ein Git-Repository als Ausgangspunkt für ein Volume verwenden .....	184
6.3	Zugriff auf Dateien im Dateisystem des Arbeitsknotens .....	187
6.3.1	HostPath-Volumes .....	188
6.3.2	Systemods mit hostPath-Volumes .....	188
6.4	Dauerhafte Speicherung .....	190
6.4.1	Eine GCE Persistent Disk in einem Pod-Volume .....	190
6.4.2	Andere Arten von Volumes mit zugrunde liegendem persistenten Speicher .....	193
6.5	Pods von der zugrunde liegenden Speichertechnologie entkoppeln .....	195
6.5.1	Persistente Volumes und Claims .....	195
6.5.2	Ein persistentes Volume erstellen .....	196
6.5.3	Mit einem Claim ein persistentes Volume beanspruchen .....	198
6.5.4	Einen Claim in einem Pod verwenden .....	200
6.5.5	Vorteile der Verwendung von persistenten Volumes und Claims ..	201
6.5.6	Persistente Volumes wiederverwenden .....	202
6.6	Persistente Volumes dynamisch bereitstellen .....	204
6.6.1	Die verfügbaren Speichertypen mit Speicherklassen definieren ..	204
6.6.2	Die Speicherklasse in einem Claim angeben .....	205
6.6.3	Dynamische Bereitstellung ohne Angabe einer Speicherklasse ...	207
6.7	Zusammenfassung .....	210
<b>7</b>	<b>Konfigurationszuordnungen und Secrets: Anwendungen konfigurieren .....</b>	<b>213</b>
7.1	Konfiguration von Anwendungen im Allgemeinen .....	213
7.2	Befehlszeilenargumente an Container übergeben .....	215
7.2.1	Den Befehl und die Argumente in Docker definieren .....	215
7.2.2	Den Befehl und die Argumente in Kubernetes überschreiben ...	217
7.3	Umgebungsvariablen für einen Container einrichten .....	219
7.3.1	Eine Umgebungsvariable in einer Containerdefinition festlegen ..	220
7.3.2	Im Wert einer Variablen auf andere Umgebungsvariablen verweisen .....	220
7.3.3	Die Nachteile hartkodierter Umgebungsvariablen .....	221
7.4	Die Konfiguration mit einer Konfigurationszuordnung entkoppeln .....	221
7.4.1	Einführung in Konfigurationszuordnungen .....	221
7.4.2	Eine Konfigurationszuordnung erstellen .....	223

7.4.3	Einen Konfigurationseintrag als Umgebungsvariable an einen Container übergeben	226
7.4.4	Alle Einträge einer Konfigurationszuordnung auf einmal als Umgebungsvariablen übergeben	227
7.4.5	Einen Konfigurationseintrag als Befehlszeilenargument übergeben	228
7.4.6	Konfigurationsdateien mithilfe eines configMap-Volumes verfügbar machen	229
7.4.7	Die Konfiguration einer Anwendung ohne Neustart ändern	235
7.5	Sensible Daten mithilfe von Geheimnissen an Container übergeben	237
7.5.1	Einführung in Geheimnisse	238
7.5.2	Das Geheimnis default-token	238
7.5.3	Ein Geheimnis erstellen	240
7.5.4	Unterschiede zwischen Konfigurationszuordnungen und Geheimnissen	241
7.5.5	Das Geheimnis in einem Pod verwenden	243
7.5.6	Geheimnisse zum Abrufen von Images	247
7.6	Zusammenfassung	248
<b>8</b>	<b>Von Anwendungen aus auf Podmetadaten und andere Ressourcen zugreifen</b>	<b>249</b>
8.1	Metadaten über die Downward-API übergeben	249
8.1.1	Die verwendbaren Metadaten	250
8.1.2	Metadaten über Umgebungsvariablen verfügbar machen	251
8.1.3	Metadaten über Dateien in einem downwardAPI-Volume übergeben	254
8.2	Kommunikation mit dem Kubernetes-API-Server	257
8.2.1	Die REST-API von Kubernetes	258
8.2.2	Von einem Pod aus mit dem API-Server kommunizieren	262
8.2.3	Botschaftercontainer zur Vereinfachung der Kommunikation mit dem API-Server	268
8.2.4	Clientbibliotheken zur Kommunikation mit dem API-Server	270
8.3	Zusammenfassung	273
<b>9</b>	<b>Deployments: Anwendungen deklarativ aktualisieren</b>	<b>275</b>
9.1	Anwendungen in Pods aktualisieren	276
9.1.1	Alte Pods löschen und anschließend durch neue ersetzen	277
9.1.2	Neue Pods starten und danach die alten löschen	277
9.2	Automatische schrittweise Aktualisierung mit einem Replikationscontroller	279
9.2.1	Die ursprüngliche Version der Anwendung ausführen	279
9.2.2	Die schrittweise Aktualisierung mit kubectl durchführen	281
9.2.3	Warum ist kubectl rolling-update veraltet?	285
9.3	Deployments zur deklarativen Verwaltung von Anwendungen	287

9.3.1	Ein Deployment erstellen	287
9.3.2	Ein Deployment aktualisieren	290
9.3.3	Eine Bereitstellung zurücknehmen	294
9.3.4	Die Rolloutrate festlegen	297
9.3.5	Den Rolloutvorgang anhalten	299
9.3.6	Das Rollout fehlerhafter Versionen verhindern	301
9.4	Zusammenfassung	306
<b>10</b>	<b>StatefulSets: Replizierte statusbehaftete Anwendungen bereitstellen</b>	<b>307</b>
10.1	Statusbehaftete Pods replizieren	307
10.1.1	Mehrere Replikate mit jeweils eigenem Speicher ausführen	308
10.1.2	Eine unveränderliche Identität für jeden Pod bereitstellen	309
10.2	Statussätze	311
10.2.1	Statussätze und Replikationssätze im Vergleich	311
10.2.2	Unveränderliche Netzwerkidentität	312
10.2.3	Eigenen beständigen Speicher für jede Podinstanz zuweisen	314
10.2.4	Garantien von Statusätzen	316
10.3	Statussätze nutzen	317
10.3.1	Die Anwendung und das Containerimage erstellen	317
10.3.2	Die Anwendung mithilfe eines Statussatzes bereitstellen	318
10.3.3	Die Pods untersuchen	323
10.4	Peers im Statussatz finden	327
10.4.1	Die Peer-Ermittlung über DNS einrichten	329
10.4.2	Einen Statussatz aktualisieren	330
10.4.3	Den Clusterdatenspeicher ausprobieren	331
10.5	Umgang mit Knotenausfällen	332
10.5.1	Die Trennung eines Knotens vom Netzwerk simulieren	332
10.5.2	Den Pod manuell löschen	334
10.6	Zusammenfassung	336
<b>Teil III: Fortgeschrittene Themen</b>		<b>337</b>
<b>11</b>	<b>Interne Mechanismen von Kubernetes</b>	<b>339</b>
11.1	Die Architektur	339
11.1.1	Die verteilte Natur der Kubernetes-Komponenten	340
11.1.2	Verwendung von etcd	343
11.1.3	Aufgaben des API-Servers	346
11.1.4	Benachrichtigungen des API-Servers über Ressourcenänderungen	348
11.1.5	Der Scheduler	350
11.1.6	Die Controller im Controller-Manager	352
11.1.7	Die Rolle des Kubelets	357
11.1.8	Die Rolle des Kubernetes-Dienstproxys	358
11.1.9	Kubernetes-Add-ons	359
11.1.10	Zusammenfassung	361

11.2	Kooperation der Controller .....	361
11.2.1	Die betroffenen Komponenten .....	361
11.2.2	Die Abfolge der Ereignisse .....	362
11.2.3	Clusterereignisse beobachten .....	364
11.3	Laufende Pods .....	365
11.4	Das Podnetzwerk .....	366
11.4.1	Anforderungen an das Netzwerk .....	366
11.4.2	Funktionsweise des Netzwerks .....	368
11.4.3	CNI .....	370
11.5	Implementierung von Diensten .....	370
11.5.1	Der Kube-Proxy .....	370
11.5.2	Iptables-Regeln .....	371
11.6	Cluster mit hoher Verfügbarkeit .....	373
11.6.1	Anwendungen hochverfügbar machen .....	373
11.6.2	Die Komponenten der Kubernetes-Steuerebene hochverfügbar machen .....	374
11.7	Zusammenfassung .....	377
<b>12</b>	<b>Sicherheit des Kubernetes-API-Servers .....</b>	<b>379</b>
12.1	Authentifizierung .....	379
12.1.1	Benutzer und Gruppen .....	380
12.1.2	Dienstkonten .....	381
12.1.3	Dienstkonten erstellen .....	382
12.1.4	Ein Dienstkonto mit einem Pod verknüpfen .....	384
12.2	Rollengestützte Zugriffssteuerung .....	386
12.2.1	Das RBAC-Autorisierungs-Plug-in .....	386
12.2.2	RBAC-Ressourcen .....	388
12.2.3	Rollen und Rollenbindungen .....	391
12.2.4	Clusterrollen und Clusterrollenbindungen .....	395
12.2.5	Standardclusterrollen und -clusterrollenbindungen .....	404
12.2.6	Berechtigungen bedachtsam gewähren .....	406
12.3	Zusammenfassung .....	407
<b>13</b>	<b>Sicherheit der Clusterknoten und des Netzwerks .....</b>	<b>409</b>
13.1	Die Namespaces des Hostknotens in einem Pod verwenden .....	409
13.1.1	Den Netzwerknnamespace des Knotens in einem Pod verwenden ..	410
13.1.2	Bindung an einen Hostport ohne Verwendung des Host-Netzwerknnamespace .....	411
13.1.3	Den PID- und den IPC-namespace des Knotens verwenden .....	413
13.2	Den Sicherheitskontext eines Containers einrichten .....	414
13.2.1	Einen Container unter einer bestimmten Benutzer-ID ausführen ..	415
13.2.2	Die Ausführung eines Containers als root verhindern .....	416
13.2.3	Pods im privilegierten Modus ausführen .....	417
13.2.4	Einem Container einzelne Kernelfähigkeiten hinzufügen .....	418



13.2.5	Fähigkeiten von einem Container entfernen .....	420
13.2.6	Prozesse am Schreiben im Dateisystem des Containers hindern ..	421
13.2.7	Gemeinsame Nutzung von Volumes durch Container mit verschiedenen Benutzer-IDs .....	422
13.3	Die Bearbeitung der Sicherheitsmerkmale in einem Pod einschränken ....	424
13.3.1	Podsicherheitsrichtlinien .....	424
13.3.2	Die Richtlinien runAsUser, fsGroup und supplementalGroups ....	427
13.3.3	Zulässige, unzulässige und Standardfähigkeiten festlegen .....	429
13.3.4	Die verwendbaren Arten von Volumes einschränken .....	430
13.3.5	Benutzern und Gruppen unterschiedliche Podsicherheitsrichtlinien zuweisen .....	431
13.4	Das Podnetzwerk isolieren .....	434
13.4.1	Die Netzwerkisolierung in einem Namespace aktivieren .....	435
13.4.2	Einzelnen Pods im Namespace die Verbindung zu einem Serverpod erlauben .....	435
13.4.3	Das Netzwerk zwischen Kubernetes-Namespaces isolieren .....	436
13.4.4	Verwendung der CIDR-Schreibweise zur Isolierung .....	438
13.4.5	Den ausgehenden Datenverkehr von Pods einschränken .....	438
13.5	Zusammenfassung .....	439
<b>14</b>	<b>Die Computerressourcen eines Pods verwalten .....</b>	<b>441</b>
14.1	Ressourcen für die Container eines Pods anfordern .....	441
14.1.1	Pods mit Ressourcenanforderungen erstellen .....	442
14.1.2	Einfluss der Ressourcenanforderungen auf die Zuteilung zu Knoten .....	443
14.1.3	Der Einfluss der CPU-Anforderungen auf die CPU-Zeitzuteilung ..	448
14.1.4	Benutzerdefinierte Ressourcen definieren und anfordern .....	448
14.2	Die verfügbaren Ressourcen für einen Container einschränken .....	449
14.2.1	Harte Grenzwerte für die von einem Container verwendeten Ressourcen festlegen .....	449
14.2.2	Überschreiten der Grenzwerte .....	451
14.2.3	Grenzwerte aus der Sicht der Anwendungen in den Containern ..	452
14.3	QoS-Klassen für Pods .....	454
14.3.1	Die QoS-Klasse eines Pods festlegen .....	454
14.3.2	Auswahl des zu beendenden Prozesses bei zu wenig Speicher ....	457
14.4	Standardanforderungen und -grenzwerte für die Pods in einem Namespace festlegen .....	458
14.4.1	Der Grenzwertbereich .....	459
14.4.2	Einen Grenzwertbereich erstellen .....	460
14.4.3	Die Grenzwerte durchsetzen .....	461
14.4.4	Standardanforderungen und -grenzwerte anwenden .....	462
14.5	Die in einem Namespace insgesamt verfügbaren Ressourcen beschränken	463
14.5.1	Ressourcenkontingente .....	463
14.5.2	Kontingente für persistenten Speicher festlegen .....	465

14.5.3	Die Höchstzahl der Objekte in einem Namespace beschränken ...	466
14.5.4	Kontingente für einzelne Podstatus und QoS-Klassen festlegen ...	467
14.6	Die Ressourcennutzung der Pods überwachen .....	468
14.6.1	Die tatsächliche Ressourcennutzung erfassen .....	468
14.6.2	Verlaufsdaten des Ressourcenverbrauchs speichern und analysieren .....	470
14.7	Zusammenfassung .....	474
<b>15</b>	<b>Automatische Skalierung von Pods und Clusterknoten .....</b>	<b>475</b>
15.1	Automatische horizontale Podskalierung .....	476
15.1.1	Der Vorgang der automatischen Skalierung .....	476
15.1.2	Skalierung auf der Grundlage der CPU-Nutzung .....	479
15.1.3	Skalierung auf der Grundlage der Speichernutzung .....	486
15.1.4	Skalierung auf der Grundlage anderer Messgrößen .....	486
15.1.5	Geeignete Messgrößen für die automatische Skalierung auswählen .....	488
15.1.6	Herunterskalieren auf null Replikate .....	489
15.2	Automatische vertikale Podskalierung .....	489
15.2.1	Ressourcenanforderungen automatisch einrichten .....	489
15.2.2	Ressourcenanforderungen von laufenden Pods ändern .....	490
15.3	Horizontale Skalierung von Clusterknoten .....	490
15.3.1	Der Cluster-Autoskalierer .....	490
15.3.2	Den Cluster-Autoskalierer aktivieren .....	492
15.3.3	Die Unterbrechung von Diensten beim Herunterskalieren des Clusters minimieren .....	493
15.4	Zusammenfassung .....	495
<b>16</b>	<b>Erweiterte Planung .....</b>	<b>497</b>
16.1	Pods mithilfe von Mängeln und Tolerierungen von bestimmten Knoten fernhalten .....	497
16.1.1	Mängel und Tolerierungen .....	498
16.1.2	Einem Knoten benutzerdefinierte Mängel hinzufügen .....	500
16.1.3	Tolerierungen zu Pods hinzufügen .....	500
16.1.4	Verwendungszwecke für Mängel und Tolerierungen .....	501
16.2	Knotenaffinität .....	502
16.2.1	Feste Knotenaffinitätsregeln aufstellen .....	503
16.2.2	Knotenprioritäten bei der Zuteilung eines Pods .....	505
16.3	Pods mit Affinitäts- und Antiaffinitätsregeln auf denselben Knoten unterbringen .....	508
16.3.1	Podaffinitätsregeln zur Bereitstellung von Pods auf demselben Knoten .....	509
16.3.2	Pods im selben Schaltschrank, in derselben Verfügbarkeitszone oder derselben geografischen Region bereitstellen .....	511
16.3.3	Präferenzen statt fester Regeln für die Podaffinität angeben .....	513

16.3.4	Pods mit Antiaffinitätsregeln voneinander getrennt halten .....	514
16.4	Zusammenfassung .....	516
<b>17</b>	<b>Best Practices für die Anwendungsentwicklung .....</b>	<b>519</b>
17.1	Das Gesamtbild .....	519
17.2	Der Lebenszyklus eines Pods .....	521
17.2.1	Beendigung und Verlegung von Anwendungen .....	521
17.2.2	Tote oder teilweise tote Pods neu bereitstellen .....	524
17.2.3	Pods in einer bestimmten Reihenfolge starten .....	525
17.2.4	Lebenszyklushooks .....	527
17.2.5	Pods herunterfahren .....	531
17.3	Die ordnungsgemäße Verarbeitung aller Clientanforderungen sicherstellen .....	535
17.3.1	Unterbrechungen von Clientverbindungen beim Hochfahren eines Pods verhindern .....	535
17.3.2	Unterbrechungen von Clientverbindungen beim Herunterfahren eines Pods verhindern .....	535
17.4	Einfache Ausführung und Handhabung von Anwendungen in Kubernetes .	540
17.4.1	Einfach zu handhabende Containerimages erstellen .....	540
17.4.2	Images sauber kennzeichnen .....	541
17.4.3	Mehrdimensionale statt eindimensionaler Labels .....	541
17.4.4	Ressourcen mit Anmerkungen beschreiben .....	542
17.4.5	Gründe für die Beendigung eines Prozesses angeben .....	542
17.4.6	Anwendungsprotokolle .....	544
17.5	Empfohlene Vorgehensweisen für Entwicklung und Tests .....	546
17.5.1	Anwendungen während der Entwicklung außerhalb von Kubernetes ausführen .....	546
17.5.2	Minikube für die Entwicklung .....	547
17.5.3	Versionssteuerung und Manifeste zur automatischen Bereitstellung von Ressourcen .....	549
17.5.4	Ksonnet als Alternative zu YAML- und JSON-Manifesten .....	549
17.5.5	Continuous Integration und Continuous Delivery (CI/CD) .....	550
17.6	Zusammenfassung .....	551
<b>18</b>	<b>Kubernetes erweitern .....</b>	<b>553</b>
18.1	Eigene API-Objekte definieren .....	553
18.1.1	Eigene Ressourcendefinitionen .....	554
18.1.2	Benutzerdefinierte Ressourcen mit benutzerdefinierten Controllern automatisieren .....	558
18.1.3	Benutzerdefinierte Objekte validieren .....	562
18.1.4	Einen benutzerdefinierten API-Server für benutzerdefinierte Objekte bereitstellen .....	562
18.2	Kubernetes mit dem Kubernetes-Dienstkatalog erweitern .....	564
18.2.1	Der Dienstkatalog .....	565

18.2.2	Der API-Server des Dienstkatalogs und der Controller-Manager ..	566
18.2.3	Dienstbroker und die API OpenServiceBroker .....	567
18.2.4	Dienste bereitstellen und nutzen .....	568
18.2.5	Aufheben der Bindung und der Bereitstellung .....	571
18.2.6	Vorteile des Dienstkatalogs .....	571
18.3	Plattformen auf der Grundlage von Kubernetes .....	572
18.3.1	Die Containerplattform Red Hat OpenShift .....	572
18.3.2	Deis Workflow und Helm .....	575
18.4	Zusammenfassung .....	578
<b>Anhang A: Verwendung von kubectl für mehrere Cluster .....</b>		<b>579</b>
A.1	Umschalten zwischen Minikube und Google Kubernetes Engine .....	579
A.1.1	Umschalten zu Minikube .....	579
A.1.2	Umschalten zu GKE .....	579
A.2	Verwendung von kubectl für mehrere Cluster oder Namespaces .....	580
A.2.1	Den Speicherort der Konfigurationsdatei festlegen .....	580
A.2.2	Der Inhalt der Konfigurationsdatei .....	580
A.2.3	Konfigurationseinträge auflisten, hinzufügen und ändern .....	581
A.2.4	Verwendung von kubectl mit verschiedenen Clustern, Benutzern und Kontexten .....	583
A.2.5	Umschalten zwischen Kontexten .....	583
A.2.6	Kontexte und Cluster auflisten .....	584
A.2.7	Kontexte und Cluster löschen .....	584
<b>Anhang B: Einen Cluster mit mehreren Knoten mit kubeadm erstellen .....</b>		<b>585</b>
B.1	Das Betriebssystem und die erforderlichen Pakete einrichten .....	585
B.1.1	Die virtuelle Maschine erstellen .....	585
B.1.2	Den Netzwerkadapter für die VM einrichten .....	586
B.1.3	Das Betriebssystem installieren .....	587
B.1.4	Docker und Kubernetes installieren .....	590
B.1.5	Die VM klonen .....	591
B.2	Den Master mit kubeadm konfigurieren .....	593
B.2.1	Ausführung der Komponenten durch kubeadm .....	594
B.3	Arbeitsknoten mit kubeadm einrichten .....	595
B.3.1	Das Containernetzwerk einrichten .....	596
B.4	Vom lokalen Computer auf den Cluster zugreifen .....	597
<b>Anhang C: Andere Containerlaufzeitumgebungen verwenden .....</b>		<b>599</b>
C.1	Docker durch rkt ersetzen .....	599
C.1.1	Kubernetes zur Verwendung von rkt einrichten .....	599
C.1.2	rkt in Minikube ausprobieren .....	600
C.2	Andere Containerlaufzeiten über die CRI verwenden .....	602
C.2.1	CRI-O .....	602
C.2.2	Anwendungen in VMs statt in Containern ausführen .....	602

<b>Anhang D: Clusterverbund</b> .....	<b>603</b>
D.1 Der Kubernetes-Clusterverbund .....	603
D.2 Die Architektur .....	604
D.3 Verbund-API-Objekte .....	605
D.3.1 Verbundversionen der Kubernetes-Ressourcen .....	605
D.3.2 Funktionsweise von Verbundressourcen .....	606
<b>Anhang E: Kubernetes-Ressourcen in diesem Buch</b> .....	<b>609</b>
<b>Index</b> .....	<b>613</b>



# Vorwort

Nachdem ich schon einige Jahre für Red Hat gearbeitet hatte, wurde ich Ende 2014 einem neuen Team namens *Cloud Enablement* zugeteilt. Unsere Aufgabe bestand darin, die Middleware-Produktpalette unseres Unternehmens auf die Containerplattform OpenShift zu übertragen, die zu diesem Zeitpunkt auf der Grundlage von Kubernetes entwickelt wurde. Damals steckte Kubernetes noch in den Kinderschuhen – noch nicht einmal Version 1.0 war veröffentlicht worden!

In unserem Team mussten wir uns mit den Interna von Kubernetes schnell vertraut machen, um unsere Software in die richtige Richtung lenken und alle Möglichkeiten ausnutzen zu können, die Kubernetes bot. Wenn wir auf ein Problem stießen, konnten wir oft schlecht unterscheiden, ob wir irgendetwas falsch machten oder ob es an einem der Bugs lag, unter denen Kubernetes in seiner Frühzeit litt.

Sowohl Kubernetes als auch meine Kenntnisse darüber haben sich seitdem erheblich weiterentwickelt. Als ich begann, damit zu arbeiten, hatten die meisten noch nie davon gehört. Heute kennt es praktisch jeder Softwareentwickler, und unter allen Möglichkeiten, um Anwendungen sowohl in der Cloud als auch in Rechenzentren am eigenen Standort auszuführen, gehört es zu denen mit der weitesten Verbreitung und dem schnellsten Wachstum.

Während der ersten Monate meiner Arbeit mit Kubernetes schrieb ich einen zweiteiligen Blogpost darüber, wie man einen JBoss-WildFly-Anwendungsservercluster in OpenShift/Kubernetes ausführt. Damals ahnte ich es noch nicht, aber dieser einfache Post führte schließlich dazu, dass der Verlag Manning mit der Bitte an mich herantrat, ein Buch über Kubernetes zu schreiben. Natürlich konnte ich ein solches Angebot nicht ablehnen, obwohl ich sicher war, dass Manning auch andere mögliche Autoren angesprochen hatte und sich letzten Endes für jemand anderen entscheiden würde.

Wie Sie sehen, geschah das nicht. Nach mehr als anderthalb Jahren Schreib- und Recherchearbeit ist das Buch nun fertig geworden. Es war ein großartiges Erlebnis. Ein Buch über ein technisches Thema zu schreiben ist die beste Möglichkeit, um die betreffende Technologie viel ausführlicher kennenzulernen, als es durch reine Anwendung möglich wäre. Da sich nicht nur meine Kenntnisse über Kubernetes erweitert haben, sondern Kubernetes selbst weiterentwickelt wurde, musste ich ständig bereits fertiggestellte Kapitel umschreiben und ergänzen. Als Perfektionist werde ich niemals absolut zufrieden mit diesem Buch sein, aber ich freue mich zu hören, dass viele Personen, die Vorabversionen über das Manning Early Access Program gelesen haben, es für einen großartigen Leitfaden zum Thema Kubernetes halten.

Mein Ziel bestand darin, den Lesern die Technologie von Kubernetes nahezubringen und den Gebrauch der Werkzeuge für eine wirkungsvolle und rationelle Entwicklung und Bereitstellung von Anwendungen in Kubernetes-Clustern vorzuführen. Die Einrichtung und Wartung eines hochverfügbaren Kubernetes-Clusters ist jedoch kein Schwerpunkt dieses Buchs, allerdings dürfte Ihnen der letzte Teil solide Kenntnisse darüber vermitteln, woraus ein solcher Cluster besteht, sodass Sie andere Quellen zu diesem Thema besser verstehen können.

Ich hoffe, dass Sie die Lektüre genießen können und das Buch Ihnen zeigt, wie Sie den größten Nutzen aus Kubernetes ziehen können.

## ■ Der Autor



Marko Lukša ist Softwareentwickler mit mehr als 20 Jahren Berufserfahrung, wobei die Palette seiner Projekte von einfachen Webanwendungen bis zu vollständigen ERP-Systemen, Frameworks und Middleware reicht. Seine ersten Programmierversuche hat er bereits 1985 im Alter von sechs Jahren auf einem ZX Spectrum gemacht, den sein Vater gebraucht für ihn gekauft hatte. In der Grundschule wurde er Landesmeister im Logo-Programmierwettbewerb und nahm an Programmierferienlagern teil, in denen er Pascal lernte. Seitdem hat er Software in einer breiten Palette von Programmiersprachen entwickelt.

In der weiterführenden Schule begann er damit, dynamische Websites zu erstellen, als das Web noch ziemlich jung war. Während seines Studiums der Informatik an der Universität von Ljubljana in Slowenien entwickelte er bei einem ortsansässigen Unternehmen Software für das Gesundheitswesen und die Telekommunikationsbranche. Schließlich begann er für Red Hat zu arbeiten. Dort entwickelte er zu Anfang eine Open-Source-Implementierung der Google App Engine API, die die Middleware JBoss von Red Hat verwendete. Außerdem arbeitete er an Projekten wie CDI/Weld, Infinispan/JBoss Data Grid u. a. mit.

Seit Ende 2014 gehört er zum Cloud-Enablement-Team von Red Hat. Zu seinen Aufgaben gehört es dabei, sich über die neuesten Entwicklungen bei Kubernetes und verwandten Technologien auf dem neuesten Stand zu halten und dafür zu sorgen, dass die Middleware des Unternehmens die Möglichkeiten von Kubernetes und OpenShift voll ausnutzt.



## ■ Danksagung zur englischsprachigen Ausgabe

Bevor ich mit dem Schreiben dieses Buches begann, hatte ich keine Vorstellung davon, wie viele Personen daran beteiligt sind, um aus einem ersten Manuskript eine fertige Veröffentlichung zu machen. Es gibt viele Menschen, denen ich Dank schulde.

Als Erstes möchte ich Erin Twohey danken, die mich gebeten hat, dieses Buch zu schreiben, und Michael Stephens, der von Anfang an volles Vertrauen darin gesetzt hat, dass ich es schaffen kann. Seine ermutigenden Worte haben mich zu Anfang stark motiviert und diese Motivation während der letzten anderthalb Jahre aufrechterhalten.

Ich möchte auch meinem ursprünglichen Entwicklungsredakteur Andrew Warren danken, der mir half, das erste Kapitel fertigzustellen, und seiner Nachfolgerin Elesha Hyde, die mit mir danach bis zum letzten Kapitel gearbeitet hat. Vielen Dank dafür, dass sie es mit mir ausgehalten haben, auch wenn der Umgang mit mir nicht ganz leicht ist und ich ziemlich oft einfach vom Radar verschwinde.

Ich möchte auch Jeanne Boyarsky danken, die als erste Lektorin meine Kapitel las und kommentierte, während ich noch daran schrieb. Jeanne und Elesha trugen erheblich dazu bei, das Buch so gut zu machen, wie es hoffentlich ist. Ohne ihre Kommentare hätte das Buch niemals so positive Bewertungen von externen Gutachtern und Lesern bekommen können.

Ich möchte auch meinem Fachlektor Antonio Magnaghi und natürlich allen externen Gutachtern danken: Al Krinker, Alessandro Campeis, Alexander Myltsev, Csaba Sari, David DiMaria, Elias Rangel, Erisk Zelenka, Fabrizio Cucci, Jared Duncan, Keith Donaldson, Michael Bright, Paolo Antinori, Peter Perlepes und Tiklu Ganguly. Ihre positive Rückmeldung hat mich durchhalten lassen, wenn ich manchmal das Gefühl hatte, mein Text sei fürchterlich geschrieben und völlig nutzlos, und ihre konstruktive Kritik hat mir geholfen, die Abschnitte zu verbessern, die ich ohne große Anstrengung flott zusammengestoppelt hatte. Vielen Dank dafür, dass Sie mich auf schwer verständliche Stellen hingewiesen und Vorschläge zur Verbesserung des Buches gemacht haben. Vielen Dank auch dafür, die richtigen Fragen zu stellen, die mir deutlich machten, dass ich ein oder zwei Dinge in der ursprünglichen Version meines Manuskripts falsch dargestellt hatte.

Ich möchte auch den Lesern danken, die Vorabversionen dieses Buches über das Early-Access-Programm von Manning (MEAP) erworben und ihre Kommentare im Onlineforum abgegeben oder mich direkt angesprochen haben, insbesondere Vimal Kansal, Paolo Patierno und Roland Huß, die einige Inkonsistenzen und andere Fehler gefunden haben. Des Weiteren möchte ich allen Manning-Mitarbeitern danken, die an der Produktion dieses Buches beteiligt waren. Bevor ich zum Schluss komme, möchte ich meinem Kollegen und Schulfreund Aleš Justin danken, der mich zu Red Hat gebracht hat, und meinen wunderbaren Kollegen im Cloud-Enablement-Team. Wäre ich nicht bei Red Hat und in diesem Team gewesen, so wäre ich nicht derjenige gewesen, der dieses Buch geschrieben hat.

Abschließend möchte ich meiner Frau und meinem Sohn danken, die während der letzten 18 Monate mehr als verständnisvoll waren und mich unterstützt haben, obwohl ich mich in mein Büro verkrochen habe, anstatt Zeit mit ihnen zu verbringen.

*Vielen Dank euch allen!*



# Über dieses Buch

Dieses Buch soll Sie zu einem kompetenten Kubernetes-Benutzer machen. Sie lernen hier praktisch alle Prinzipien kennen, die Sie beherrschen müssen, um Anwendungen in einer Kubernetes-Umgebung zu entwickeln und auszuführen.

Bevor es mit Kubernetes losgeht, erhalten Sie einen Überblick über Containertechnologien wie Docker und das Erstellen von Containern, damit Sie den weiteren Ausführungen auch dann folgen können, wenn Sie damit noch nicht gearbeitet haben. Danach werden Sie Schritt für Schritt in alles eingeführt, was Sie über Kubernetes wissen müssen, von den Grundprinzipien zu den verborgenen Mechanismen.

## ■ Zielgruppe

Dieses Buch richtet sich hauptsächlich an Anwendungsentwickler, bietet aber auch einen Überblick über die Verwaltung von Anwendungen. Es ist für alle gedacht, die sich für die Ausführung und Verwaltung von Containeranwendungen auf mehr als einem einzigen Server interessieren.

Sowohl Anfänger als auch erfahrene Softwareentwickler, die etwas über Containertechnologien lernen möchten, erhalten hier die notwendigen Kenntnisse, um ihre Anwendungen in einer Kubernetes-Umgebung zu entwickeln und in Containern auszuführen.

Vorkenntnisse in Containertechnologien und Kubernetes sind nicht erforderlich. Die Erklärungen in diesem Buch bauen aufeinander auf, und es wird kein Beispielcode verwendet, der nur für Experten verständlich wäre.

Leser sollten allerdings Grundkenntnisse in Programmierung, Computernetzwerken, einfachen Linux-Befehlen und Standardprotokollen wie HTTP mitbringen.

## ■ Der Aufbau dieses Buches

Dieses Buch besteht aus 18 Kapiteln, die in drei Teile gegliedert sind.

Teil 1 gibt eine kurze Einführung in Docker und Kubernetes. Sie erfahren hier, wie Sie einen Kubernetes-Cluster einrichten und darin eine einfache Anwendung ausführen. Dieser Teil umfasst nur zwei Kapitel:

- Kapitel 1 erklärt, was Kubernetes ist, wie es entstand und wie es hilft, die heutigen Probleme der Verwaltung und Skalierung von Anwendungen zu lösen.
- Kapitel 2 enthält eine praktische Anleitung, um ein Containerimage zu erstellen und in einem Kubernetes-Cluster auszuführen. Es erklärt auch, wie Sie lokale Kubernetes-Cluster mit einem Knoten und richtige Cluster mit mehreren Knoten in der Cloud ausführen.

Teil 2 stellt die Grundprinzipien vor, mit denen Sie vertraut sein müssen, um Anwendungen in Kubernetes auszuführen. Er besteht aus folgenden Kapiteln:

- Kapitel 3 stellt die Grundbausteine von Kubernetes vor – die Pods – und erklärt, wie Pods und andere Kubernetes-Objekte mithilfe von Labels geordnet werden können.
- Kapitel 4 erklärt, wie Kubernetes Anwendungen durch den automatischen Neustart von Containern in funktionsfähigem Zustand hält. Hier erfahren Sie auch, wie Sie verwaltete Pods ausführen, horizontal skalieren, gegen den Ausfall von Clusterknoten absichern und zu vorherbestimmten Zeitpunkten oder regelmäßig ausführen.
- Kapitel 5 zeigt, wie Pods ihre Dienste für Clients im und außerhalb des Clusters verfügbar machen und wie Pods im Cluster Dienste innerhalb oder außerhalb des Clusters entdecken und nutzen können.
- Kapitel 6 erklärt, wie mehrere Container im selben Pod Dateien gemeinsam nutzen können und wie Sie persistenten Speicher verwalten und für die Pods zugänglich machen.
- Kapitel 7 zeigt, wie Sie Konfigurationsdaten und sensible Informationen, z. B. Anmeldeinformationen, an Anwendungen innerhalb der Pods übergeben.
- Kapitel 8 beschreibt, wie Anwendungen Informationen über die Kubernetes-Umgebung beziehen, in der sie laufen, und wie sie mit Kubernetes kommunizieren, um den Status des Clusters zu ändern.
- Kapitel 9 gibt eine Einführung in das Prinzip von Deployments und erklärt, wie Sie Anwendungen in einer Kubernetes-Umgebung ordnungsgemäß ausführen und aktualisieren.
- Kapitel 10 stellt eine Möglichkeit zur Ausführung statusbehafteter Anwendungen vor, die gewöhnlich eine stabile Identität benötigen und ihren Status erhalten müssen.

In Teil 3 geht es um die internen Mechanismen von Kubernetes-Clustern. Es werden hier nicht nur einige neue Konzepte eingeführt, sondern auch alle Funktionsprinzipien, die Sie in den ersten beiden Teilen gelernt haben, von einer höheren Warte aus untersucht. Dieser Teil umfasst folgende Kapitel:

- Kapitel 11 stößt unter die Oberfläche von Kubernetes vor und beschreibt die Komponenten, aus denen ein Kubernetes-Cluster besteht, und ihre Funktionsweise. Außerdem wird hier erklärt, wie Pods über das Netzwerk kommunizieren und wie Dienste die Last auf mehrere Pods verteilen.

- Kapitel 12 erklärt, wie Sie Ihren Kubernetes-API-Server und den Cluster mithilfe von Authentifizierung und Autorisierung sicherer gestalten können.
- Kapitel 13 zeigt, wie Pods auf die Ressourcen des Knotens zugreifen und wie ein Clusteradministrator sie daran hindern kann.
- Kapitel 14 beschreibt, wie Sie den Verbrauch von Computerressourcen durch die einzelne Anwendung einschränken, die garantierte Dienstqualität der Anwendungen festlegen, die Ressourcennutzung überwachen und Benutzer daran hindern, zu viele Ressourcen zu verbrauchen.
- Kapitel 15 erklärt, wie Sie Kubernetes einrichten, um die Anzahl der Replikat einer Anwendung automatisch skalieren zu lassen, und wie Sie die Größe des Clusters erhöhen, wenn die vorhandenen Clusterknoten keine weiteren Anwendungen mehr aufnehmen können.
- Kapitel 16 zeigt, wie Sie dafür sorgen, dass Pods bestimmten Knoten zugeteilt oder nicht zugeteilt werden. Außerdem erfahren Sie, wie Sie Pods zusammen oder getrennt zuteilen können.
- Kapitel 17 beschreibt, wie Sie Ihre Anwendungen auf clustergerechte Weise entwickeln sollten. Sie erhalten dabei auch einige Hinweise dazu, wie Sie bei der Entwicklung und dem Testen vorgehen sollten, um Störungen zu vermeiden.
- Kapitel 18 zeigt, wie Sie Kubernetes mit eigenen Objekten erweitern können und wie andere dies bereits getan und damit professionelle Anwendungsplattformen erstellt haben.

Bei der Lektüre werden Sie nicht nur die einzelnen Bestandteile von Kubernetes kennenlernen, sondern auch Ihre Kenntnisse des Befehlszeilentools `kubectl` nach und nach erweitern.

## ■ Der Code

Dieses Buch enthält nicht viel Quellcode, aber dafür eine Menge Manifeste für Kubernetes-Ressourcen im YAML-Format sowie Shellbefehle und deren Ausgabe. All diese Elemente sind in nichtproportionaler Schrift dargestellt, um sie vom normalen Fließtext abzuheben.

Die Shellbefehle sind gewöhnlich in fetter nichtproportionaler Schrift angegeben, um sie von der Ausgabe zu unterscheiden. Manchmal sind jedoch nur die wichtigsten Teile eines Befehls oder auch einige besondere Teile der Ausgabe fett hervorgehoben. Die Ausgabe wurde meistens umformatiert, sodass sie in eine Buchzeile passt. Da das Kubernetes-Befehlszeilenwerkzeug `kubectl` ständig weiterentwickelt wird, kann es sein, dass sich die Ausgabe neuerer Versionen von dem unterscheidet, was Sie in diesem Buch sehen. Die Listings enthalten oft auch Anmerkungen, die die wichtigsten Teile hervorheben und erklären.

Alle Beispiele in diesem Buch wurden mit Kubernetes 1.8 in der Google Kubernetes Engine und in einem lokalen Minikube-Cluster getestet. Der vollständige Quellcode und die YAML-Manifeste sind auf <https://github.com/luksa/kubernetes-in-action> zu finden.

## ■ Das Forum zum Buch

Manning Publications, der Herausgeber der Originalausgabe, unterhält ein Forum (in englischer Sprache), in dem Sie das Buch kommentieren, fachliche Fragen stellen und Hilfe sowohl vom Autor als auch von anderen Lesern erhalten können. Dieses Forum finden Sie auf <https://forums.manning.com/forums/kubernetes-in-action>. Um mehr über die Manning-Foren und die Verhaltensregeln dafür zu erfahren, schauen Sie auf <https://forums.manning.com/forums/about> nach.

Manning möchte damit eine Möglichkeit für den Austausch zwischen Lesern und zwischen Leser und Autor geben. Der Autor ist dabei in keiner Form zu irgendeiner garantierten Form der Beteiligung verpflichtet, da er alle seine Beiträge freiwillig (und unbezahlt) leistet. Wir raten Ihnen, anspruchsvolle Fragen zu stellen, falls sein Interesse nachlassen sollte. Das Forum und die Archive früherer Diskussionen sind auf der Website von Manning so lange zugänglich, wie die Originalausgabe dieses Buches in Druck ist.

## ■ Sonstige Onlinequellen

Informationen über Kubernetes sind auch an folgenden Orten zu finden:

- Auf der Kubernetes-Website <https://kubernetes.io>
- Im Kubernetes-Blog auf <http://blog.kubernetes.io>, in dem regelmäßig interessante Informationen erscheinen
- Im Slack-Kanal der Kubernetes-Community auf <http://slack.k8s.io>
- In den YouTube-Kanälen von Kubernetes und der Cloud Native Computing Foundation:
- [https://www.youtube.com/channel/UCZ2bu0qutTOM0tHYa\\_jkIwg](https://www.youtube.com/channel/UCZ2bu0qutTOM0tHYa_jkIwg)
- <https://www.youtube.com/channel/UCvqbFHwN-nwaIWPjPUKpvTA>

Um mehr über einzelne Themen zu erfahren oder auch um selbst zu Kubernetes beitragen zu können, wenden Sie sich an die Kubernetes-Interessengruppen (Special Interest Groups, SIGs) auf [https://github.com/kubernetes/kubernetes/wiki/Special-Interest-Groups-\(SIGs\)](https://github.com/kubernetes/kubernetes/wiki/Special-Interest-Groups-(SIGs)).



Da es sich bei Kubernetes um Open-Source-Software handelt, enthält auch der Kubernetes-Quellcode einen Schatz an Informationen. Sie finden ihn in <https://github.com/kubernetes/kubernetes> und verwandten Repositories.