

# Kubernetes

Das Praxisbuch für Entwickler und DevOps-Teams

» Hier geht's  
direkt  
zum Buch

# DAS VORWORT

# Kapitel 1

## Vorwort

*»Be water my friend.«*

*Bruce Lee*

Vielleicht kennen Sie das Interview mit Bruce Lee, aus dem dieses Zitat stammt. Im Kern seiner Aussage geht es um die Anpassungsfähigkeit des Wassers. Was Bruce Lee meinte, ist, sich im Kampf auf sein Gegenüber einzustellen. Sich anzupassen, wenn es notwendig ist. Offen zu sein, um adäquat auf die Umwelt und die sich verändernden Gegebenheiten reagieren zu können. Dieser Satz begleitet mich schon eine ganze Weile. Nicht nur, weil ich Kampfsport ausübe, sondern vor allem, weil diese Metapher auch in der IT so passend ist.

Als ich 2011 meine Ausbildung zum Fachinformatiker Systemintegration angefangen habe, wusste ich noch gar nicht, welche Welt sich mir öffnen würde. Ich habe das Handwerk IT von Grund auf gelernt. Ich kann mich noch gut an die ersten Projekte erinnern. Für eine Anwaltskanzlei haben wir ein Netzwerk verlegt. Die Löcher selbst gebohrt, die Kabel gezogen und die Netzwerkstecker gecrimpt. Oder ich habe selbst für ein mittelständisches Unternehmen einen neuen Server zusammengebaut, installiert und in einem Serverraum vor Ort verkabelt und eingerichtet.

Wenn Sie heute einen Server brauchen, dann klicken Sie auf LAUNCH im Frontend Ihres Cloud-Anbieters. Auf magische Weise wird eine virtuelle Maschine (VM) in irgendeinem Hochsicherheitsrechenzentrum für Sie freigeschaltet. Die Welt ist ständig im Wandel, und so ist es auch mit der Technologie. Durch die Abstraktion wird das Benutzen immer einfacher, doch das bringt nicht nur Vorteile mit sich.

2017 bin ich nach dem Studium zum ersten Mal in Berührung mit Cloud, DevOps, Docker und Kubernetes gekommen, und seither lässt mich das Thema nicht mehr los. Ich erinnere mich noch an die Tage während des Studiums, als ich mich gefragt habe: »Wie kommt meine Software eigentlich zum Kunden?«

In der Ausbildung und im Studium habe ich programmieren gelernt. Erst Java, dann C, dann C++. Ich habe gelernt, was Softwareengineering ist, wie ich UML-Diagramme erstelle und wie ich in *Assembler* maschinennah entwickeln kann. Doch eines hat mir immer gefehlt: Wie kommt die Software am Ende zum Kunden? Der Betrieb war nie wirklich Teil meiner Ausbildung gewesen. Es geht mir um Fragen wie:

- ▶ Wie wird die Software gebaut und verpackt?
- ▶ Wie wird die Software ausgeliefert?
- ▶ Was passiert, wenn die Software mal nicht funktioniert?

Meistens habe ich meine eigene Entwicklung auf meinem Rechner oder in der IDE laufen lassen. Auf meine Fragen zum letzten Puzzlestück konnte mir lange Zeit niemand eine zufriedenstellende Antwort geben.

Ich habe dual studiert, und in den Praxisphasen habe ich in Abteilungen gearbeitet, die Softwarekonzepte erstellten. Meine Kollegen haben sich Gedanken darüber gemacht, welche Anforderungen das Business hat, wie das in Software gegossen werden könnte, und haben seitenweise Spezifikationen geschrieben und UML-Diagramme gemalt. Die Software wurde dann von Partnerunternehmen on- und offshore entwickelt, was meist suboptimal funktioniert hat. Es waren klassische Wasserfall-Projekte. Auch hier habe ich den Prozess der Softwareentwicklung nie von Anfang bis Ende verstanden. Ich dachte immer: »Da fehlt doch was. Irgendwie passt das noch nicht.« Und manchmal dachte ich: »Vielleicht bin ich dafür einfach zu blöd.«

Heute weiß ich, dass ich nicht zu blöd bin und mir einfach nur das letzte kleine Puzzlestück gefehlt hat. Nach dem Studium kam ich zum ersten Mal in den Kontakt mit moderner agiler Softwareentwicklung. Ein Team konzipiert, entwickelt und liefert die Software aus, und ein anderes macht dafür den Betrieb. Das war auch das erste Mal, dass ich die Begriffe *Cloud*, *Docker* und *Kubernetes* gehört habe. Die agile Softwareentwicklung war mir im Studium schon begegnet, doch die ganzen Tools waren mir neu. Meine Welt wurde komplett auf den Kopf gestellt.

Virtuelle Maschinen kannte ich noch von der Ausbildung, und ich wusste auch, dass eine Cloud mehr als nur ein Speicherdienst wie Dropbox oder OneDrive ist. Heute bin ich unter anderem zertifizierter *AWS Certified Solutions Architect – Professional*, und während ich über meine Vergangenheit schreibe, muss ich schmunzeln. Die Welt der IT ist so viel größer, als ich sie mir je hätte vorstellen können, und ich habe wirklich meine Leidenschaft entdeckt.

Seitdem ich mit der Cloud, Docker und Kubernetes arbeite, fühlt es sich für mich rund an. Mir ist jetzt klar, wie moderner Softwarebetrieb funktioniert und was dafür notwendig ist. Mit diesem Buch möchte ich Sie an meinen Erkenntnissen teilhaben lassen.

2017 habe ich auf der Plattform Udemy einen Onlinekurs über den Einstieg in Docker veröffentlicht. Ich hatte das Ziel, den Einstieg einfacher zu machen. Damit die Teilnehmenden ein Gefühl für Docker bekommen, indem sie selbst ins Tun kommen.

Mir hat das nämlich bis dato im dualen Studium gefehlt. Ich bin Praktiker und lerne am besten, wenn ich etwas mit meinen eigenen Händen mache.

Auch dieses Buch ist so geschrieben, dass Sie einen schnellen Einstieg in das Thema Kubernetes bekommen. Sie werden selbst ein Cluster aufbauen und erste Services deployen. Ich nehme Sie mit auf die Reise und werde Sie strukturiert an das Thema heranführen. Ich liebe es, wenn es einfach ist. Kompliziert wird es ganz von allein und schnell genug, deswegen werden Sie Schritt für Schritt alles lernen, was wichtig ist, um die Software fit für Kubernetes zu machen.

### Danksagung

Mein Sohn Levi Ace: Du zeigst mir jeden Tag, was im Leben wirklich wichtig ist.

Meine Frau Nicole: Danke, dass du mir immer den Rücken freihältst, auch wenn ich bis spät in die Nacht am Buch schreibe.

Mein bester Freund und Geschäftspartner Fabian: Danke für die gemeinsame Reise seit über einer Dekade.

## 1.1 Gliederung

Lassen Sie mich kurz darauf eingehen, was Sie auf den nächsten Seiten erwarten wird. Sie sind schon mittendrin im ersten Kapitel. Ich möchte Sie dort abholen, wo Sie stehen, und Sie einführen in das, was Sie erwartet.

Kapitel 2 und Kapitel 3 habe ich für Sie wie ein Tutorial konzipiert. Jeder Abschnitt baut auf dem anderen auf. Sie werden Grundlagen und Konzepte lernen und dann sehr schnell ins Tun kommen. Nach diesen Kapiteln sind Sie bereit für Kubernetes und können dann einzelne Themen vertiefen. Denn ab Kapitel 4 werden wir uns einzelne Aspekte genauer ansehen. Diese Kapitel können Sie in der Reihenfolge lesen, in der Sie es gerade brauchen.

### Hinweis

Manche Abschnitte in Kapitel 2 und Kapitel 3 sind sehr gut zum Nachlesen geeignet. Wenn Sie in einem bestimmten Abschnitt das Gefühl haben: »Gerade nicht der richtige Moment«, dann lesen Sie es gerne quer. Dadurch füttern Sie Ihr Unterbewusstsein und kommen bei Bedarf wieder darauf zurück, um genauer nachzulesen.



In Kapitel 4 nehme ich Sie mit zum Thema *Infrastructure as Code*. Sie lernen YAML und den Unterschied zwischen deklarativer und imperativer Arbeit kennen.



### Hinweis

Sie werden in Kapitel 3 schon mit YAML und Infrastructure as Code arbeiten. Sie können dann gerne zu Kapitel 4 springen und einen kleinen Exkurs machen, bevor Sie mit Kapitel 3 weitermachen.

In Kapitel 5 werden wir in fortgeschrittenere Konzepte und Objekte einsteigen.

Kapitel 6 widmet sich dem Thema Storage- und Stateful-Anwendungen. Was machen Sie mit Anwendungen, die wie Datenbanken einen State haben? Welche Arten von Storage gibt es in Kubernetes, und wie können Sie Ihre Daten am besten verwalten.

Ein großes Thema in der IT ist Security und Governance. Dazu werde ich Sie in Kapitel 7 in Themen wie User- und Rechteverwaltung, Pod Security und Kubernetes Policies einführen. Sie werden die Grundlagen kennenlernen, um Ihre Anwendung auf die Produktion vorbereiten zu können.

In Kapitel 8 werden Sie alles lernen, was Sie wissen müssen, um Ihre Anwendung »ready for Production« zu machen: Ressourcen-Management, Health Checks und Skalierung für Ihre Anwendungen.

Last, but not least werden Sie in Kapitel 9 den Kubernetes-Paketmanager Helm kennenlernen. Schon in früheren Kapiteln werden Sie Helm einsetzen, um in Beispielen fertige Anwendungen zu deployen. Helm wird Ihnen die Welt als Entwickler sehr einfach machen und Ihnen dabei helfen, Ihre Anwendung fit für mehrere Umgebungen zu machen.

### 1.1.1 Was Sie schon wissen sollten

Kubernetes ist eine Software, die auf anderen Themen wie beispielsweise Container aufbaut, die selbst Bücher füllen. Aus diesem Grund gibt es einige Voraussetzungen, die in diesem Buch erwartet werden, damit Sie schnell an das Thema Kubernetes herangeführt werden und wir uns nicht in den Details verstricken.

Das Tool Kubernetes ist ein Container-Management-System. Grundlegendes Wissen über Container ist daher notwendig. Fragen wie die folgenden sollten Sie beantworten können:

- ▶ Was ist ein Container?
- ▶ Wie ist ein Container aufgebaut?
- ▶ Wie kann ich Container-Images bauen und meine Software in einen Container bringen?
- ▶ Wie kann ich Container starten und stoppen?
- ▶ Wie funktioniert das Container-Tool auf meinem Rechner?

Ich nutze Docker Desktop als Basis, baue Container mit Dockerfiles und werde mit Ihnen in Abschnitt 2.4 ein Testcluster mit Minikube als Container aufbauen. Je nach Betriebssystem können Sie natürlich auch andere Tools wie Podman verwenden. Sie sind auch nicht auf Minikube angewiesen und können andere Testcluster verwenden, wenn Sie sich damit auskennen.

#### Hinweis

Wenn Sie mit einem Unternehmensrechner arbeiten, dann kann es sein, dass Sie eine Lizenz für Docker Desktop benötigen. Sind Sie sich unsicher, dann fragen Sie am besten noch mal nach, oder nutzen Sie Ihren privaten Computer.



Die Container, die in diesem Buch verwendet werden, basieren alle auf Linux-Images. Entsprechend ist es von Vorteil, wenn Sie sich mit den Basics von Linux auskennen. Auch sollten Sie sich gut auf der Kommandozeile wie *Bash* bzw. *Shell* bewegen können. Wenn Sie schon mal Bash-Skripts geschrieben haben, dann wird das, was wir in diesem Buch verwenden, ein Kinderspiel für Sie werden. Falls Sie die Beispiele auf Basis eines Windows-Betriebssystems durchführen, sollten Sie mit der PowerShell umgehen können. Doch auch hier werde ich Ihnen die wichtigsten Befehle bereitstellen.

Wir werden *Command Line Interface (CLI)* Tools wie *kubectl* und *Minikube* verwenden. Das sind Programme, die auf der Kommandozeile ausgeführt werden, um beispielsweise Kubernetes zu bedienen. Hier werde ich Sie Schritt für Schritt heranzuführen, doch Sie werden sich schneller zurechtfinden, wenn Sie mit CLI-Tools schon gearbeitet haben.

Generell ist dieses Buch jedoch für Anfänger geeignet. Sie werden Kubernetes von der Pike auf lernen, und ich werde versuchen, Sie so gut es geht da abzuholen, wo Sie stehen. Das bedeutet, selbst wenn Sie kaum Erfahrung mit den oben genannten Tools mitbringen, werden Sie dieses Buch durcharbeiten können. An manchen Stellen wird es für Sie dann vielleicht von Vorteil sein, wenn Sie das Buch wegpacken und sich zusätzliche Lerninhalte zu Docker und Co schnappen.

### 1.1.2 Was Sie lernen werden

Wie bereits erwähnt, richtet sich dieses Buch an Entwickler und DevOps-Engineers, die sich in Kubernetes einarbeiten wollen – egal ob Sie erst neulich von Kubernetes gehört haben oder sich schon länger damit befassen. Nach diesem Buch haben Sie das Handwerkszeug, um Ihre Anwendungen für Kubernetes zu entwickeln und zu betreiben. Sie werden resiliente, skalierbare und zuverlässige Infrastrukturen aufbauen können. Ihre Anwendungen werden durch Self-Healing und Loadbalancing reif für produktive Umgebungen sein.

Sie werden in diesem Buch nicht oder nur am Rande lernen, wie Sie ein Kubernetes-Cluster installieren oder administrieren können. Sie werden jedoch lernen, wie Sie Anwendungen in einem Cluster betreiben können und was Sie als Entwickler oder Entwicklerin dabei beachten müssen. Sie lernen, ein Kubernetes-Cluster zu bedienen und zu steuern. Dafür werden Sie ein Testcluster auf Ihrem Computer auf Basis von Minikube installieren. Das wird Ihnen helfen, auf einer Testumgebung alles ausprobieren zu können.

### 1.1.3 Wichtig zu wissen

In den fortgeschrittenen Kapiteln werden Sie tiefer in die Eigenheiten von Kubernetes einsteigen. Für die Beispiele verwende ich einfache Applikationen als Beispiel. Wenn Sie direkt versuchen, die Beispiele mit Ihren eigenen Anwendungen durchzuspielen, dann ist der Lerneffekt deutlich höher, und Sie können das Gelernte viel besser in Ihrer täglichen Arbeit implementieren.

Vielleicht kennen Sie das auch: Sie wollen einen Italienurlaub machen und lernen zuhause mit einer Sprachapp die wichtigsten Begriffe wie »Guten Tag«, »Auf Wiedersehen« und »Apfel«. Auch kleine Sätze wie »Einen Kaffee, bitte« und »Ich würde gerne bezahlen«.

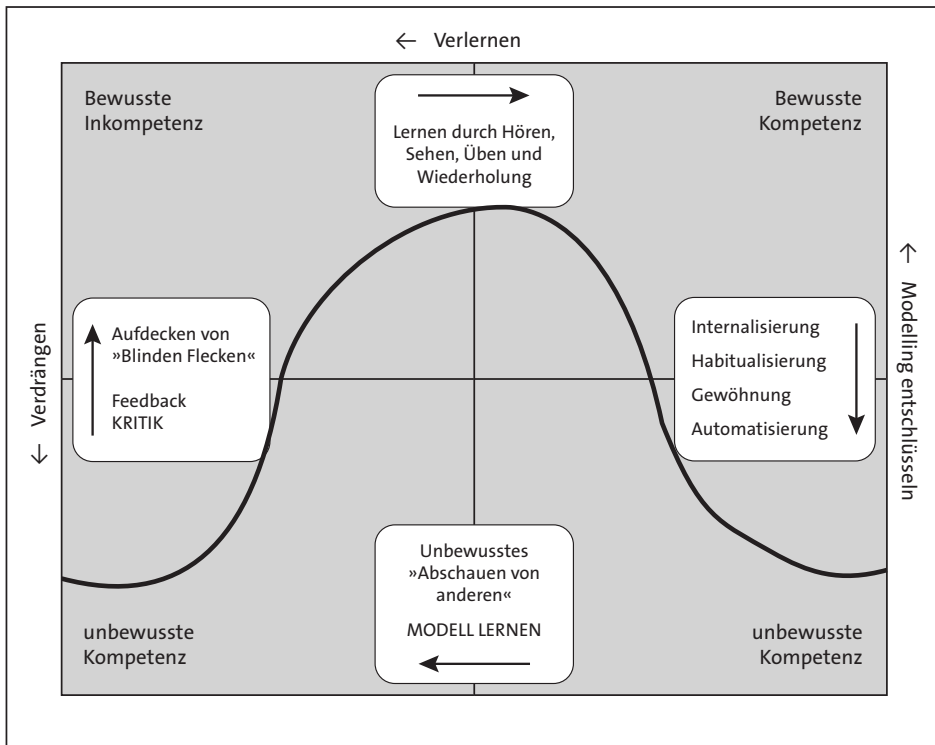
Dann ist es so weit. Sie sind im Urlaub angekommen und betreten das erste Café. Der Kellner fragt Sie, was Sie haben möchten, und auf einmal ist all das Gelernte vergessen. Ihnen kommt kein Wort über die Lippen.

Die Gedächtnisforschung zeigt: Der Abruf einer neuen Fähigkeit gelingt dann am besten, wenn die Umstände möglichst denen entsprechen, unter denen die neuronalen Verknüpfungen stattgefunden haben. Wenn wir also zuhause am Schreibtisch mit der App lernen, dann fällt uns das Abrufen der Informationen mit der App am leichtesten. In einer neuen Situation wie beispielsweise im italienischen Café sind die Umstände anders, und wir können die Informationen nicht mehr abrufen.

Aus diesem Grund ist es wichtig, dass Sie jedes Kapitel entweder direkt oder zusätzlich mit eigenen Projekten ausprobieren. Dadurch wird sich alles besser vernetzen, und Sie können die Inhalte deutlich besser anwenden. Haben Sie Vertrauen in den Prozess dieses Buchs. Am Ende wird sich alles fügen, und Sie werden Kubernetes in Ihrem Umfeld erfolgreich einsetzen können. Bis dahin müssen wir noch einige Schritte gehen, doch ich werde Sie dabei begleiten.

Dazu möchte ich Ihnen noch ein Modell des kanadischen Psychologen Albert Bandura vorstellen (siehe Abbildung 1.1). Vielleicht kennen Sie es sogar schon, denn es ist ein bekanntes Modell, wie Lernen bei uns Menschen funktioniert. Bandura teilt das menschliche Lernen in vier Phasen ein:

1. unbewusste Inkompetenz
2. bewusste Inkompetenz
3. bewusste Kompetenz
4. unbewusste Kompetenz



**Abbildung 1.1** Vier Phase des Lernens nach Albert Bandura

Denken Sie doch dabei an Ihren Autoführerschein. Können Sie sich erinnern, wie es für Sie war, als Sie zum ersten Mal hinterm Steuer saßen? Oder das erste Mal das Auto an der Ampel abgewürgt haben? Ihnen wird auf einmal bewusst, dass Ihnen eine Fähigkeit fehlt. Sie wollen Auto fahren, doch Ihnen wird auch klar, dass das kein einfacher Weg wird, denn der einzige Weg von bewusster Inkompetenz zu bewusster Kompetenz ist harte Arbeit. Sie lernen und üben. Immer und immer wieder.

Nach einer Zeit können Sie nun Auto fahren. Doch in Ihrem Kopf ist immer wieder die kleine Stimme, die sagt: »Jetzt schalten«, »Schulterblick und blinken«. Sie müssen mit vollem Bewusstsein und kognitiver Anstrengung fahren. Erst nach vielen Stunden Auto fahren wird die Tätigkeit allmählich einfacher, und Sie entwickeln einen Automatismus. Heute sitze ich im Auto, höre meinem Sohn zuliebe Kindermusik und singe lautstark mit. Das Fahren selbst ist unbewusst und einfach geworden.



Mit dem Kauf dieses Buchs haben Sie die erste Phase schon hinter sich gelassen. Ihnen ist bewusst, dass Sie lernen müssen. Der Übergang von Phase zwei zu drei ist der härteste und häufig mit viel Frustration verbunden. Das Buch wird Ihnen den Übergang so leicht wie möglich machen, damit Sie Freude beim Umsetzen haben und so mit Leichtigkeit zum Kubernetes-Profi werden. Ich wünsche Ihnen viel Erfolg und Spaß beim Lesen, Lernen und Spielen.

## 1.2 Materialien zum Buch

Auf der Webseite zu diesem Buch stehen folgende Materialien für Sie zum Download bereit:

### ► Listings

Gehen Sie auf [www.rheinwerk-verlag.de/9663](http://www.rheinwerk-verlag.de/9663). Klicken Sie auf den Reiter MATERIALIEN. Sie sehen die herunterladbaren Dateien samt einer Kurzbeschreibung des Dateiinhalts. Klicken Sie auf den Button HERUNTERLADEN, um den Download zu starten. Je nach Größe der Datei (und Ihrer Internetverbindung) kann es einige Zeit dauern, bis der Download abgeschlossen ist.