

# Terraform

Das Praxisbuch für DevOps-Teams und Administratoren

» Hier geht's  
direkt  
zum Buch

# DAS VORWORT

# Kapitel 1

## Einleitung

Sehr geehrte Leserin, sehr geehrter Leser,

wir freuen uns, dass wir Sie auf eine Reise in Richtung *Infrastructure as Code* und Terraform mitnehmen dürfen. Wir wünschen Ihnen einen angenehmen Flug, am Ziel erwartet uns herrliches Wetter.

Spaß beiseite, schön, dass Sie den Umgang mit Terraform lernen wollen. *Infrastructure as Code* ist aus unserem Alltag nicht mehr wegzudenken. Nehmen Sie sich Zeit, sich über die Vor- und Nachteile des Arbeitens mit Terraform zu informieren. Finden Sie den passenden Weg für sich, Ihr Team und Ihr Unternehmen, um Ihre Infrastruktur zu organisieren und zu verwalten.

Wir wünschen gutes Gelingen!

**Tim Beermann**

**Johannes Kastl**

**Eike Waldt**

**Christian Rost**

**Thorsten Schifferdecker**

### 1.1 Gliederung

Die nächsten Abschnitte verraten Ihnen, was Sie wissen sollten, bevor Sie mit diesem Buch loslegen können. Sie erfahren, was Sie lernen werden und was nicht zum Ziel dieses Buchs gehört.

Kapitel 2 führt in grundlegende Prinzipien und Denkweisen ein und unterfüttert das beliebte *Buzzword Bingo* (*DevOps! IaC! Cloud!*) mit solidem Hintergrundwissen. Ziel ist es nicht, Sie mit langatmigen Ausschweifungen zu langweilen. Dennoch sollen Sie verstehen, wieso Terraform in den letzten Jahren so eine steile Karriere hingelegt hat und womit das zusammenhängt.

Kapitel 3 zeigt die Installation von Terraform und bringt Ihnen die ersten Befehle zum Arbeiten mit Terraform näher.

In Kapitel 4 dürfen Sie dann die ersten Ressourcen bei einem Cloud-Anbieter aufbauen. Das Beispielszenario wird für die Cloud-Anbieter AWS, Microsoft Azure und Google Cloud Platform sowie die Technologien OpenStack, VMware und libvirt gezeigt.

Kapitel 5 behandelt das Gehirn von Terraform, den sogenannten *State*, und zeigt, was Sie darüber wissen müssen und was Sie damit machen können.

In Kapitel 6 zeigen wir Ihnen Eingabevariablen, Ausgabewerte und dergleichen mehr. Sie lernen verschiedene Wege kennen, auf denen Sie Variablen Werte zuweisen können, und erfahren, wann welcher Weg sinnvoll oder schwierig umzusetzen ist.

Die von Terraform verwendete Beschreibungssprache *HCL (HashiCorp Configuration Language)* wird in Kapitel 7 detailliert behandelt. Sie erfahren (fast) alles über Anweisungen, Funktionen, Module und derlei Dinge mehr. Auch das Thema Versionsverwaltung wird hier erstmalig eingehender besprochen.

Kapitel 8 befasst sich mit den anstehenden Aufgaben nach dem ersten erfolgreichen Aufbauen Ihrer Infrastruktur. Dies umfasst Themen wie Aktualisierungen für Terraform und dessen Provider, den Lebenszyklus erstellter Ressourcen und wie Sie Datenverlust vermeiden.

Die Organisation und Struktur Ihres Terraform-Codes ist eines der Themen, die in Kapitel 9 abgehandelt werden. Auch die Handhabung von Zugangsdaten bei der Nutzung einer *Versionsverwaltung* oder eines *Remote Backend* werden thematisiert.

In Kapitel 10 wollen wir Ihnen ein in sich abgeschlossenes Deployment vorstellen, das als Beispiel für viele Anwendungsszenarien steht.

Kapitel 11 wagt den Blick über den Tellerrand und schneidet Themen an, die den Umfang dieses Buchs sprengen würden. Dazu gehören Automatisierung rund um Terraform, die *Terraform Cloud* und *Terraform Enterprise* sowie Werkzeuge aus dem Terraform-Umfeld, wie z. B. *Terragrunt* oder *Timon*.

Eine Kommandozeilenreferenz zu Terraform, all seinen Befehlen und deren Parametern finden Sie in Kapitel 12.

### 1.1.1 Was Sie schon wissen sollten

Terraform dient zur Verwaltung von Infrastrukturressourcen. In diesem Buch werden Ihnen Ressourcen wie virtuelle Maschinen, Netzwerke, Blockstorage und Ähnliches begegnen. Sie werden es außerdem mit Begriffen wie Webserver, Datenbanken, Loadbalancer und Backups zu tun haben. Sie müssen nicht alle diese Ressourcen und Begriffe im Schlaf definieren und beschreiben können, es reicht, wenn Sie ein Grundverständnis dieser Ressourcen mitbringen. Grundlegende Kenntnisse zu Virtualisierung (Was ist eine virtuelle Maschine?), Netzwerktechnik (Was ist eine IP-Adresse?) und Administration (Wie melden Sie sich per SSH an?) sollten vorhanden sein.

Kapitel 2 führt in Terraform, dessen Nutzung und verwandte Bereiche wie *Cloud Computing*, *DevOps* und *Infrastructure as Code* ein und stellt Terraform in den größeren Kontext.

Terraform ist ein Kommandozeilenprogramm ohne grafische Oberfläche. Sie sollten daher mit dem Arbeiten auf einer Kommandozeile vertraut sein. Im Idealfall kennen Sie sich auf der Kommandozeile des Betriebssystems aus, auf dem Sie Terraform installieren und nutzen wollen. Die Installation selbst wird in Kapitel 3 für verschiedene Betriebssysteme beschrieben.

Die meisten Beispiele werden auf einem Linux-System in einer bash-Shell gezeigt. Linux-Kenntnisse sind nicht notwendig, hier aber durchaus hilfreich. Sie sollten jedoch auch ohne Vorerfahrung verstehen können, was in den Beispielen gezeigt wird.

### Anmerkungen

In den Beispielen werden Sie häufiger IP-Adressen sehen. Diese sind lediglich beispielhaft gewählt und werden eventuell nicht bei Ihnen funktionieren.

Die Kapitel führen abwechselnd in femininer und maskuliner Schreibweise durch den Inhalt, um den Lesefluss nicht zu stören.

## 1.1.2 Was Sie in diesem Buch lernen und nicht lernen werden

Ziel dieses Buchs ist es, Ihnen die Idee hinter Terraform und *Infrastructure as Code* nahezubringen und Ihnen grundlegende Kenntnisse in Terraform und Terraform-Code zu vermitteln. Hierzu werden »in verschiedenen Clouds«, d. h. bei verschiedenen Cloud-Anbietern, Ressourcen aufgebaut. Sie müssen nicht alle Beispiele durchexerzieren. Es wäre jedoch von Vorteil, wenn Sie Zugang zu einem der vorgestellten Cloud-Anbieter bzw. einer der Technologien haben. Selbstverständlich müssen Sie auch nicht alle Dienste im Portfolio aller Cloud-Anbieter kennen, erklären und verwenden können. Dieses Buch beschränkt sich auf grundlegende Ressourcen.

Sollten Sie noch unerfahren sein, was das Arbeiten mit Cloud-Anbietern angeht, empfehlen wir die hauseigenen Tutorials der Cloud-Anbieter. Hier lernen Sie die Grundlagen zum Arbeiten mit dem Cloud-Anbieter Ihrer Wahl. Wenn Sie diese kennen, können Sie sich voll und ganz darauf konzentrieren, die bekannten Ressourcen mithilfe von Terraform aufzubauen, zu verwalten und gegebenenfalls zu entfernen. Für Anwender, die für den Cloud-Provider ein Benutzerkonto ihrer Organisation verwenden, mag es einfacher sein, sich einen neuen Benutzerzugang zu erstellen, der nicht mit ihrer Unternehmensorganisation verbandelt ist. Zum einen reicht das kostenlose Probekonto aus, um die Beispiele in diesem Buch durchzuspielen, zum anderen besteht weniger Risiko, existierende Infrastruktur während Ihrer Experimente zu beeinträchtigen. Darüber hinaus kann es sein, dass Dinge wie *Tokens* oder technische Benutzer mit den Rechten Ihres Unternehmenszugangs nicht erstellt werden dürfen. Im Zweifel hilft hier Ihr unternehmensinterner Ansprechpartner für den Zugang zu Ihrem Cloud-Anbieter.



### Zertifizierung zu Terraform

Ein Hinweis in eigener Sache: Die Firma HashiCorp bietet eine Zertifizierung zu Terraform an. Wir weisen darauf hin, dass dieses Buch *nicht* für die Vorbereitung auf die Zertifizierung ausreicht. Sollte eine Zertifizierung geplant sein, verwenden Sie bitte zusätzlich das Lernmaterial, das HashiCorp selbst frei zugänglich bereitstellt. Lesen Sie die gesamte Dokumentation zur Terraform-Sprache *HCL*. Es werden zum Beispiel detaillierte Fragen zu Funktionen oder auch zur *Terraform Cloud* gestellt.

Dieses Buch hat den Anspruch, Neulingen und Einsteigerinnen die Grundlagen des Arbeitens mit Terraform beizubringen. Sie werden an die Hand genommen und durch die ersten Beispiele geführt. Sie erhalten solide Grundkenntnisse in Terraform wie dem *State* und in der Terraform-eigenen *HCL*-Auszeichnungssprache und den darin enthaltenen Funktionen. Sie erhalten Erfahrungswerte zu Dingen wie Ordnerstrukturen und Dateinamenskonventionen, zur Modularisierung des Terraform-Codes und zu den weiteren Schritten nach dem berühmten ersten Schritt.

Dieses Buch erhebt keinen Anspruch auf Vollständigkeit. Es ist schlicht nicht das Ziel, alle verfügbaren Konstellationen, Parameter und Sonderfälle bis ins kleinste Detail zu beschreiben. Sie erhalten ein wertvolles Rüstzeug, mit dem Sie gut vorbereitet in Ihre ersten Terraform-Projekte gehen können.

Aufgrund der Fortentwicklung von Terraform können Unterschiede im Funktionsumfang zwischen der im Buch beschriebenen Terraform-Version und der bei Erscheinen dieses Buchs aktuellen Version nicht vermieden werden. Im Zweifelsfall gilt die offizielle Dokumentation passend zu Ihrer eingesetzten Terraform-Version.

## 1.2 Voraussetzungen für das Arbeiten mit Terraform

Im Laufe des Buchs werden Sie anhand simpler Beispielszenarios lernen, eine virtuelle Maschine mit öffentlicher IP-Adresse zu erstellen, die per SSH erreichbar ist. Sie werden dieses Beispiel in mehreren Clouds nachvollziehen, einschließlich aller zusätzlich benötigten Ressourcen. Dies umfasst Dinge wie Sicherheitsgruppen, Netzwerke, Subnetze und SSH-Schlüsselpaare. Ziel ist es, dass Sie bei Ihren ersten Lernversuchen in der Cloud tatsächlich Ressourcen erstellen, die nicht nur theoretisch funktionieren, sondern auch direkt genutzt werden können. Wie beschrieben, ist es nicht notwendig, dass Sie Zugang zu allen beschriebenen Cloud-Anbietern oder Technologien wie OpenStack haben. Sie sollten aber in der Lage sein, wenigstens eines der Beispiele selbst umzusetzen. Bitte beachten Sie, dass je nach Cloud-Anbieter und Zugang Kosten für das Aufbauen dieser Ressourcen entstehen können. Bei Erstellung dieses Buchs waren die Ressourcen von den kostenlosen Kontingenten der Cloud-An-

bieter abgedeckt. Hierfür kann jedoch keine Gewähr übernommen werden. Sprechen Sie dazu Ihren unternehmensinternen Ansprechpartner an (sofern Sie sich nicht als Privatperson fortbilden).

Um mit Terraform arbeiten zu können, benötigen Sie natürlich Zugriff auf ein System, auf dem Terraform installiert ist. Als Arbeitsumgebung eignet sich Windows, macOS oder auch ein Linux-System. Auch weitere unixoide Betriebssysteme wie FreeBSD, OpenBSD oder Solaris werden unterstützt. Die Faustregel lautet: Solange die Programmiersprache Go auf dem System unterstützt wird, lässt sich Terraform mit hoher Wahrscheinlichkeit nutzen. Mehr zur Installation von Terraform finden Sie in Kapitel 3.

Da Terraform ein Kommandozeilenprogramm ist, benötigen Sie nicht unbedingt eine grafische Bedienoberfläche auf dem System, auf dem Terraform installiert ist. Sie können Terraform zum Ausprobieren auf einer virtuellen Maschine mit Linux ausführen. Sollten Sie auf die Weboberflächen eines Cloud-Anbieters zugreifen müssen, können Sie dies auch auf einem anderen Gerät erledigen.

### 1.3 Zu den Autoren dieses Werks

Tim Beermann ist seit frühester Jugend begeisterter Linux-Administrator. Während seiner Laufbahn hat er ein ERP-System mitentwickelt und Serverlandschaften sowie Cloud-Umgebungen betreut. Heute beschäftigt er sich überwiegend mit Automatisierung, Supercomputern, Cloud- und Containertechnologien. Sein umfangreiches Wissen über Automatisierung teilt er in Schulungen zu den Themen Terraform, Ansible, Elastic Stack und SUSE Linux.

Johannes Kastl ist seit frühen SUSE-Tagen begeisterter Linux-Nutzer. Er war mehrere Jahre in der Administration einer großen privaten Cloud-Landschaft tätig, unter anderem als Administrator, technischer Product Owner, Scrum Master und Teamkoordinator. Seine Tätigkeitsschwerpunkte liegen auf der Systemadministration sowie dem System- und Konfigurationsmanagement mit z. B. Ansible, Puppet oder Chef. Dazu kommt Automatisierung sowohl in Richtung CI/CD als auch in Richtung *Infrastructure as Code* und Kubernetes. Der passionierte Git-Benutzer ist Freund agiler Methoden und hat mehrfach mitgeholfen, das Arbeiten mit Scrum oder Kanban in Teams einzuführen. Er gibt sein Wissen gern in Schulungen und Blogbeiträgen weiter und ist als Linux-Trainer und Consultant tätig.

Christian Rost begleitet und verantwortet große und kleine Projekte im Container- und Kubernetes-Umfeld, plant und baut Logging- und Monitoring-Lösungen mit Elastic Stack und steht für das eine oder andere ownCloud-Projekt zur Verfügung. Mit Terraform automatisiert er die hierfür nötigen IT-Landschaften sowohl *on-premise* als auch bei verschiedenen Cloud-Anbietern.

Thorsten Schifferdecker ist seit mehr als 20 Jahren begeisterter Linux-Nutzer und in der IT als Architekt tätig. Er hat Kunden bei der Projektierung und Aufbauphase von komplexen Infrastrukturen im Cloud- und Containerbereich – wie Kubernetes – begleitet und verantwortet. Sein Wissen stellt er neben dem *DevOps*-Projektalltag auch immer wieder gern als Trainer zur Verfügung.

Eike Waldt hat während seiner Laufbahn viele SUSE- und SAP-Projekte begleitet und geleitet. Innerhalb dieser Projekte entwarf er komplexe Mechanismen zur Verwaltung von gewachsenen Landschaften und strukturierte gewachsene Codesammlungen mittels *DevOps* und *CI*. Aktuell beschäftigt er sich mit hochverfügbaren Pacemaker-Clustern im SAP-Umfeld und automatisiert deren Bereitstellung unter anderem mit Terraform.

### Danksagungen

Die Autoren bedanken sich bei Herrn Christian Berendt, Geschäftsführer der OSISM GmbH, und Herrn Ralph Dehner, Geschäftsführer der B1 Systems GmbH.

Großer Dank gilt Christoph Meister vom Rheinwerk Verlag, der uns bei den Arbeiten an diesem Buch unterstützt und geholfen hat.

► Tim Beermann dankt

Ramona. Weil du mein persönlicher Glücksbringer bist.

► Johannes Kastl dankt und grüßt

Tine. Du bist das Wichtigste.

Meinen beiden Kindern Mathilda und Rufus, die mich mit Freude und Stolz erfüllen. Meinen Eltern und Geschwistern und Schwiegereltern und der ganzen Großfamilie, weil ihr so ein toller Haufen seid.

Fritz Reichwald.

Dieses Buch ist, wie sollte es anders sein, Jean-Luc Picard gewidmet.

► Eike Waldt dankt

Anni, Detlef, Ulrike, Sönke, dem Punkrock.

► Thorsten Schifferdecker dankt

Tanja, Neele, Noah, Maria, David, Melanie und Robert.

► Christian Rost dankt

Annika. Weil du der Deckel zu meinem Topf bist.

Josefine, unserem kleinen Sonnenschein.

Meinen Eltern für die Geduld in den letzten 33 Jahren, den besten Brüdern, die man sich wünschen kann, und meinem Neffen Liam, der mir hoffentlich bald alle seine Tricks beim Angeln verrät.

- ▶ Dank gilt allen, die an Open-Source-Werkzeugen mitarbeiten und dafür sorgen, dass unglaublich viele nützliche Programme und Hilfsmittel frei zugänglich sind. Stellvertretend für viele andere Tools seien hier besonders genannt:  
awk, ansible, diff, draw.io, find, git, GitHub, grep, hedgedoc, markdownlint, pandoc, sed, terraform, vim, Visual Studio Code, xargs