Best practise für Terraform in der Praxis - terraform-in-der-praxis.de

# Terraform in der Praxis

Da die Cloud-Infrastruktur für große und kleine Unternehmen immer wichtiger wird, sind verschiedene Tools entstanden, die Unternehmen bei der Verwaltung ihrer schnell wachsenden Cloudumgebung unterstützen u.a. Terraform. Die Bereitstellung und Verwaltung von Cloud-Ressourcen ist schneller und einfacher als je zuvor dank Infrastructure as Code (IaC). Mit IaC gehören langwierige manuelle Konfigurationen und einmalige Skripte der Vergangenheit an. Stattdessen verwalten Sie die Infrastruktur mit Code auf die gleiche Weise wie Anwendungen und Dienste. Diese Infrastruktur kann alles sein, von Servern und Datenbanken bis hin zu Netzwerken, Kubernetes-Clustern und ganzen Anwendungsstacks.

Eines der wichtigsten Cloud-Tools, das in den letzten zehn Jahren entwickelt wurde, ist [Terraform von HashiCorp](https://www.terraform.io/). Es ist ein beliebtes Multi-Cloud-IaC-Framework. Wenn Sie Ihre Cloudumgebung verwalten, besteht die Möglichkeit, dass Sie Terraform nutzen … aber tun Sie dies auf die richtige Weise? In diesem Artikel stelle ich Ihnen einige Best Practices von Terraform aus unserer Praxis vor.

### Was ist Terraform?

Terraform ist ein von HashiCorp entwickeltes Tool, das eine Abstraktionsschicht zum Beschreiben und Bereitstellen verschiedener Arten von Cloud-Infrastrukturen bereitstellt. Es funktioniert auf allen großen Cloud- Anbietern, einschließlich AWS, Google Cloud, Azure, IBM Cloud und sogar lokalen Cloud-Frameworks wie VMWare. Es verwendet einen deklarativen Ansatz, d. h. Sie definieren, wie die Infrastruktur aussehen soll, und nicht die Schritte, um dieses Ergebnis zu erreichen. Um Änderungen an der Infrastruktur vorzunehmen, wie z. B. das Hinzufügen weiterer Instanzen mit denselben Konfigurationen, definieren Sie einfach die Änderungen in der Vorlage und Terraform erledigt den Rest.

Ein weiterer Vorteil von Terraform ist, dass es modular aufgebaut ist. Das macht es für Teams einfacher, die Infrastruktur mit nur wenigen Zeilen Code bereitzustellen und zu skalieren. Durch die Änderung einiger weniger Variablen passen Sie z.B. Netzwerke, Storagekonfiguration und anderes an.

Obwohl Terraform fast überall in der Cloud-Infrastrukturlandschaft verwendet wird, gibt es immer noch viele Implementierungen, die die Leistungsfähigkeit von Terraform nicht optimal nutzen. Dieser Artikel soll zehn Best Practices zur Verbesserung Ihrer Terraform-Implementierung hervorheben. Wenn Sie sich an diese Empfehlungen halten, verbessern Sie Ihre Terraform-Projekte erheblich, sowohl im Hinblick auf die Gesamtqualität als auch auf die Geschwindigkeit der Implementierung.

### Beiträge

Ich bitte um euer Feedback um dieses Dokument ständig zu aktualisieren. Wenn neue Ideen ausgereift genug sind, werden diese überprüft und hier implementiert. Wenn Sie an bestimmten Themen interessiert sind, oder mögliche neue **Inhalte** beisteuern können [öffnen Sie ein Issue](https://github.com/AlexanderWiechert/terraform-in-der-praxis/issues).

Das Buch ist hier kostenlos erhältlich - <https://www.terraform-in-der-praxis.de/>. Gestartet wurde das Projekt von Anton Babenko. Übersetzt ins Deutsche und erweitert von [Alexander Wiechert](https://github.com/AlexanderWiechert).

### Brauchen Sie Hilfe?

Wenn Sie Hilfe oder kommerzielle Unterstützung für Terraform, AWS oder Jenkins CI benötigen, senden Sie eine E-Mail an [info@elastic2ls.com](mailto:info@elastic2ls.com) oder besuche sie unsere Webseite [www.elastic2ls.com](https://www.elastic2ls.com/)

### Lizenz

Dieses Werk ist unter der Apache 2-Lizenz lizenziert. Ausführliche Informationen finden Sie unter LICENCE. Copyright © 2021 Alexander Wiechert.

### Sponsoring

Wenn Sie das Projekt fördern wollen, [spenden](https://www.paypal.com/paypalme/AlexanderWiechert) Sie uns doch bitte einen kleinen Betrag