

Terraform

Automatisierte Provisionierung von Infrastruktur

Betriebsteams müssen sich aktuell mit der eigenen Infrastruktur und/oder mehreren Clouds parallel auseinandersetzen. Dies stellt viele Unternehmen vor große Herausforderungen.

Terraform ist ein Infrastructure as Code OpenSource-Tool, mit dem sich diese Herausforderungen meistern lassen. Es hilft bei der effizienten Verwaltung und Automatisierung der Infrastruktur, unabhängig von der zugrunde liegenden Plattform oder dem Cloud-Anbieter.

Terraform basiert auf einer eigenen deklarativen Konfigurationssprache, der Terraform Language. Mit deren Hilfe wird die gewünschte Cloud- oder lokale Infrastruktur beschrieben. Anschließend wird ein Plan zum Erreichen dieses gewünschten Zustands erstellt und dieser zur Bereitstellung der Infrastruktur ausgeführt. Terraform kann auf genauso effizientem Wege Komponenten anpassen oder löschen.

In diesem Terraform Kurs lernen Sie die Architektur und Funktionsweise von Terraform kennen. Ziel ist die effiziente Nutzung der Terraform Language sowie die optimale Strukturierung des Code in Form von Modulen. Dabei bringen wir Ihnen die Beispiele anhand praktischer Übungen näher. Lernen Sie in diesem Terraform Training eines der derzeit beliebtesten verfügbaren OpenSource-Tools zur Automatisierung kennen!

Kursinhalt

- Was ist Terraform?
- Was ist Infrastructure as Code?
- Architektur von Terraform
- Abgrenzung zu Ansible, Puppet und anderen IaC-Tools
- Aufbau von Terraform Konfigurationsdateien
- Ressourcen & Data Sources
- Provider & Provisioner
- Verschiedene Arten von Variablen
- Konditionelle Ausdrücke & Templating
- Module & Workspaces
- Interne Terraform Funktionen
- Beispiele anhand von AWS und OpenStack

E-Book Das ausführliche deutschsprachige digitale Unterlagenpaket, bestehend aus PDF und E-Book, ist im Kurspreis enthalten.

Zielgruppe

Dieses Terraform Training richtet sich an alle, die einen Überblick über Features und Funktionsweise von Terraform erhalten möchten und sich für die technischen Aspekte der Implementierung interessieren.

Voraussetzungen

Grundlegendes Know-how in den Bereichen Netzwerkkonzepte und mehrschichtige Architekturen wird für dieses Terraform Training vorausgesetzt. Einfache Cloud Computing-Kenntnisse sind hilfreich, aber keine Voraussetzung.

Dieser Kurs im Web



Alle tagesaktuellen Informationen und Möglichkeiten zur Bestellung finden Sie unter dem folgenden Link: www.experteach.de/go/TERR

Vormerkung

Sie können auf unserer Website einen Platz kostenlos und unverbindlich für 7 Tage reservieren. Dies geht auch telefonisch unter 06074 4868-0.

Garantierte Kurstermine

Für Ihre Planungssicherheit bieten wir stets eine große Auswahl garantierter Kurstermine an.

Ihr Kurs maßgeschneidert

Diesen Kurs können wir für Ihr Projekt exakt an Ihre Anforderungen anpassen.

Stand 03.06.2025

| Training | Preise zzgl. MwSt. | |
|-------------------------------|---------------------|------------------|
| Termine in Deutschland | 3 Tage | € 2.195,- |
| Termine in Österreich | 3 Tage | € 2.195,- |
| Online Training | 3 Tage | € 2.195,- |
| Termin/Kursort | Kurssprache Deutsch | |
| 11.06.-13.06.25 | ONOnline | 11.02.-13.02.26 |
| 20.08.-22.08.25 | HYFrankfurt | 11.02.-13.02.26 |
| 20.08.-22.08.25 | HYOnline | 08.04.-10.04.26 |
| 30.09.-02.10.25 | HYOnline | 08.04.-10.04.26 |
| 30.09.-02.10.25 | HYWien | 27.05.-29.05.26 |
| 15.12.-17.12.25 | HYHamburg | 27.05.-29.05.26 |
| 15.12.-17.12.25 | HYOnline | HYOnline |



Inhaltsverzeichnis

Terraform – Automatisierte Provisionierung von Infrastruktur

- 1 Einführung zu IaC - Tools**
 - 1.1 Aktuelle Herausforderungen im Betrieb
 - 1.2 Infrastructure as Code (IaC)
 - 1.2.1 Deklarativer Ansatz
 - 1.3 Abgrenzung der Tools
 - 1.3.1 Ansible
 - 1.3.2 Puppet und Chef
 - 1.3.3 AWS CloudFormation
 - 1.4 Terraform
 - 1.4.1 Terraform Open Source, Cloud & Enterprise
- 2 Installation & Grundlagen**
 - 2.1 Installation
 - 2.1.1 Terraform auf Windows installieren
 - 2.1.2 Terraform auf Ubuntu installieren
 - 2.2 Terraform Konfigurations-Dateien
 - 2.2.1 Ressourcen
 - 2.3 Provider
 - 2.3.1 Provider installieren
 - 2.4 Die wichtigsten Terraform Befehle
 - 2.4.1 Vorgehensweise
 - 2.4.2 Init & Validate
 - 2.4.3 Plan
 - 2.4.4 Apply
 - 2.4.5 Destroy
 - 2.5 Provisioner
 - 2.5.1 Provisioner remote-exec
 - 2.5.2 Provisioner file
- 3 Variablen & Referenzen**
 - 3.1 Überblick
 - 3.2 Input Variablen
 - 3.2.1 Werte-Übergabe an Input-Variablen
 - 3.3 Output Values
 - 3.3.1 Nutzen von Output Values
 - 3.4 Local Values
 - 3.5 Data Sources
 - 3.5.1 Komplexeres Beispiel für eine Data Source
- 4 Terraform State**
 - 4.1 Das Terraform Statefile
 - 4.1.1 Informationen vom State abfragen
 - 4.2 Refresh
 - 4.3 Import
 - 4.4 Statefile Remote
 - 4.4.1 Terraform Backend Konfig
 - 4.5 Infos von einem anderen Remote State
- 5 Schleifen, Bedingungen & Templates**
 - 5.1 Schleifen
 - 5.1.1 Loops mit count
 - 5.1.2 Loops mit for_each
 - 5.1.3 for_each im dynamic Block
 - 5.1.4 Loops mit for
 - 5.2 Bedingungen
 - 5.2.1 if-else Statements
 - 5.2.2 Abhängigkeiten mit depends_on
 - 5.2.3 Lifecycle - Pre & Postconditions
 - 5.2.4 weitere Lifecycle Argumente
 - 5.3 Templating
- 6 Module & Workspaces**
 - 6.1 Ordner- und Dateistruktur
 - 6.2 Strukturierung über Module
 - 6.2.1 Ordnerstruktur mit Environments
 - 6.3 Module im Detail
 - 6.3.1 Moduldeklaration und Inputs
 - 6.3.2 Modul Outputs
 - 6.3.3 Provider im Modul
 - 6.3.4 Schleifen im Modul
 - 6.4 Module Remote
 - 6.4.1 Modul Sources
 - 6.5 CLI Workspaces
 - 6.5.1 Workspaces State
 - 6.5.2 Code Beispiele bei Nutzung von Workspaces
 - 6.5.3 Module & Workspaces in Kombination
- A Abkürzungsverzeichnis**
- B Befehle**

