

Terraform Grundlagen



Was ist Terraform

- **Infrastruktur as Code (IaC):** Terraform ist ein Open-Source-Werkzeug für die Erstellung, Verwaltung und Automatisierung von Cloud-Infrastruktur mithilfe von Code.
- **Deklarative Konfiguration:** Terraform verwendet eine deklarative Sprache (HCL - HashiCorp Configuration Language), um die gewünschte Infrastruktur zu beschreiben, anstatt eine imperative Programmierung.
- **Plattformunabhängig:** Es unterstützt verschiedene Cloud-Anbieter wie AWS, Azure, Google Cloud, und mehr, sowie On-Premises-Infrastruktur.
- **Parallele Ausführung:** Terraform ermöglicht die parallele Erstellung und Verwaltung von Ressourcen, was die Effizienz erhöht.

WHEN THEY TELL YOU

THE HCL IS THE REAL PROGRAMMING LANGUAGE

- **Zustandsverwaltung:** Es erstellt eine Zustands- oder State-Datei, um den aktuellen Status der Infrastruktur festzuhalten und Änderungen nachzuvollziehen.
- **Modularität:** Terraform-Konfigurationen sind modular und wiederverwendbar, was die Skalierbarkeit und Wartbarkeit erleichtert.
- **Versionskontrolle:** Terraform-Konfigurationen können in Versionskontrollsystemen wie Git gespeichert und verwaltet werden.
- **Community und Ökosystem:** Es gibt eine aktive Terraform-Community und eine Vielzahl von Modulen, die von der Community erstellt wurden, um die Konfiguration zu erleichtern.
- **Kostenkontrolle und Compliance:** Terraform kann bei der Optimierung von Ressourcen und Einhaltung von Sicherheits- und Compliance-Richtlinien helfen.

Ressourcen

- Ressourcen sind die grundlegenden Bausteine in Terraform.
- Sie repräsentieren Ressourcen, die in der Cloud-Umgebung erstellt oder verwaltet werden sollen.
- Beispiele für Ressourcen sind EC2-Instanzen, S3-Buckets, RDS-Datenbanken usw.
- In Terraform werden Ressourcen in der Regel in einem Terraform-Konfigurationsfile (`.tf`) definiert.

Provider

- Ein Provider in Terraform ist eine Konfiguration, die die Verbindung zu einer bestimmten Cloud-Plattform herstellt.
- Jede Cloud-Plattform (z.B. AWS, Azure, Google Cloud) hat ihren eigenen Terraform-Provider.
- Der Provider definiert die Zugangsdaten, die für die Interaktion mit der Cloud-Plattform erforderlich sind.
- Ein Beispiel für die Konfiguration eines AWS-Providers in Terraform:

```
provider "aws" {  
  region = "us-east-1"  
  access_key = "Ihr_Access_Key"  
  secret_key = "Ihr_Secret_Key"  
}
```

Data Sources

- Data Sources ermöglichen es Terraform, Informationen aus der Cloud-Plattform abzurufen.
- Im Gegensatz zu Ressourcen werden mit Data Sources keine neuen Ressourcen erstellt.
- Data Sources sind nützlich, um Informationen über bereits vorhandene Ressourcen zu erhalten, z.B. Informationen über eine vorhandene EC2-Instanz.


```
data "aws_instance" "example" {  
    instance_id = "i-0123456789abcdef0"  
}  
  
output "instance_private_ip" {  
    value = data.aws_instance.example.private_ip  
}
```

Locals

- Locals sind benutzerdefinierte Variablen und Berechnungen, die in Terraform-Dateien definiert werden.
- Sie verbessern die Lesbarkeit und Wartbarkeit von Terraform-Konfigurationsdateien, indem sie häufig verwendete Werte speichern.
- Hier ist ein einfaches Beispiel für die Verwendung von Locals:

```
locals {  
  region = "us-east-1"  
  instance_type = "t2.micro"  
}  
  
resource "aws_instance" "example" {  
  ami           = "ami-0c55b159cbfafa1f0"  
  instance_type = local.instance_type  
  availability_zone = "${local.region}a"  
}
```

Variables

- **Variablen in Terraform** sind Container für Werte, die in einer Terraform-Konfiguration wiederverwendet werden können. Sie bieten Flexibilität und Konfigurierbarkeit, ohne die Notwendigkeit, harte Werte direkt in den Konfigurationsdateien festzulegen.
- Variablen können verwendet werden, um **Konfigurationen parametrisierbar** zu machen. Zum Beispiel können wir eine Variable für die AWS-Region erstellen, um leicht zwischen verschiedenen Regionen zu wechseln, ohne die Konfigurationsdatei zu ändern.
- Wir können **verschiedene Arten von Variablen** in Terraform definieren, einschließlich Strings, Zahlen, Listen und Maps. Dies ermöglicht es, verschiedene Datentypen in den Konfigurationen zu verwenden.
- Variablen können in Terraform auf **verschiedene Arten bereitgestellt** werden, einschließlich durch die Verwendung von Umgebungsvariablen, Variablen-Dateien oder direkt in der Konfigurationsdatei.

```
variable "region" {  
    description = "AWS Region"  
    type        = string  
    default     = "us-east-1"  
}  
  
resource "aws_instance" "example" {  
    ami           = "ami-0c55b159cbfafa1f0"  
    instance_type = "t2.micro"  
    availability_zone = "${var.region}a"  
}
```

Outputs

- **Ausgaben in Terraform** sind eine Möglichkeit, Werte aus der Terraform-Konfiguration zu extrahieren und anzuzeigen, nachdem sie erstellt wurden.
- Sie sind nützlich, um **Informationen über erstellte Ressourcen zugänglich zu machen**, z.B. die öffentliche IP-Adresse einer EC2-Instanz oder die DNS-Adresse eines Load Balancers.
- Ausgaben können in einer Terraform-Konfiguration definiert werden und sind für Benutzer und andere Teile der Infrastruktur zugänglich.
- Nach der Anwendung der Terraform-Konfiguration können wir die Ausgabe mit dem Befehl `terraform output` anzeigen, um die öffentliche IP-Adresse der erstellten Instanz anzuzeigen.
- Ein Beispiel für die Verwendung von Ausgaben:

```
resource "aws_instance" "example" {  
    ami          = "ami-0c55b159cbfafa1f0"  
    instance_type = "t2.micro"  
}  
  
output "public_ip" {  
    value = aws_instance.example.public_ip  
}
```

Projekt Struktur

- **All .tf Dateien werden beim Ausführen gemerget**, das heißt das Aufteilen auf mehrere Dateien dient nur der Übersicht
- Folgende Projekt Struktur ist jedoch üblich und wird auch so für Module verwendet:
 - **main.tf**: Die Hauptdatei, hier werden alle Terraform Ressourcen deklariert.
 - **versions.tf**: Die Konfiguration der verwendeten Terraform Provider und Terraform Version.
 - **variables.tf**: Hier werden alle Terraform Variables definiert.
 - **outputs.tf**: Hier werden alle Terraform Outputs definiert.