

## TRF-103: Terraform AWS - Essentials (OpenSource Edition)

This course delivers you the essentials of Infrastructure as Code (IaC) with Terraform practically. You learn the features of Terraform in a Hands-On approach. Each feature we introduce will be used with an AWS Service so you can see how it works in production. The open-source version of Terraform will be used so that you can get started immediately. Some AWS knowledge is recommended but not required. Each AWS service in this course will be introduced in a short practical manner. After attending this training, you can have a good understanding of the dos and don'ts of Terraform. You gained valuable experience using AWS Services with Terraform, including production-ready scenarios.

**Structure:** 50% theory, 50% hands on lab exercises.

**Target audience:** developers, system administrators and DevOps professionals who want to learn the concepts and usage of Terraform in AWS cloud environments.

**Prerequisites:** Basic AWS cloud knowledge

**Duration:** 3 days

### Detailed Course Outline:

#### Module 1: Overview

- Challenges in Infrastructure and Projects
- What is Terraform?
- What is Idempotence?
- Open-Source vs. Enterprise Option in Terraform
- Advantages with Terraform 1.x

#### Module 2: Installation and first steps

- Pre-reqs, AWS-Account, Credentials
- Setup / Install Terraform
- Connection to AWS – get to know Providers
- Basic-File structure for Terraform Projects (Best Practice from Projects incl. gitignore)
- Write Code for your first AWS-Resource (simple EC2 Instance with a Webserver deployment)
- Make the fundamental IaC-loop of creating, updating and deleting resources
- Add some more Attributes to a resource (like Labels)
- How to manage config-drift with Terraform
- What is a State-File and why it's important?

#### Module 3: Starting a Real-World AWS deployment with a VPC

- Fundamental Architecture in AWS
- Create your Virtual Private Cloud
- Create Subnets (public/private)
- Create an Internet-Gateway
- Create a Route Table
- Test everything with an EC2-Instance deployment

#### Module 4: Go on with IaaS - EC2 Instances with Terraform

- Create an EC2-Instance
- Create necessary Security Groups
- Create Security Groups with dynamic Blocks
- Usage of Instance-Attributes like "user\_data" for customization.

- Get more than one copy of a Resource
- Create Detailed-Monitoring with CloudWatch
- Create a CloudWatch-Alarm for a Threshold
- Create an AutoScaling-Group for the EC2-Instances
- Create an Application Load-Balancer for the Instance-Pool (TargetGroup)

### **Module 5: Terraform Variables**

- Create a more dynamic terraform deployments with Variables
- Variables-Types (Environment, Files, Commandline)
- Get to know the fundamental types: number, string, boolean
- More advanced structures: list, maps, objects, tuples
- Input and Output Variables
- Refactor the Solution with Variables for dynamic deployments

### **Module 6: Datasources and Terraform**

- The purpose of a Datasource
- Use Datasources for EC2-AMI
- Use Datasources for VPC-Metadata

### **Module 7: AWS S3 with Terraform**

- Create S3 Buckets
- Create Bucket-Policies
- Create Bucket-Policies with Template-Functions
- Upload Data to Buckets
- Host static websites with S3

### **Module 8: Terraform Remote-State Files with S3**

- Working with Remote-State-Files
- Using State files in a Remote-Bucket (S3)

### **Module 9: AWS IAM with Terraform**

- Overview of IAM in AWS
- How to create Users
- How to create custom Policies
- How to make Identity-Roles/Resource-Roles
- How to assume a Role

### **Module 10: AWS Route53 with Terraform**

- Advantages of Route53 for DNS
- Create new DNS Entries for a Zone
- Use Datasources for DNS-Zones
- Chain your static website hosting with a friendly DNS Name

### **Module 11: AWS Relational Database Services (RDS) with Terraform**

- Advantages of Managed Databases
- General prereqs. for RDS deployment (Security Groups, VPC, DB-Type, Backup)
- Create your RDS-Instance with Terraform (MySQL)

### **Module 12: Use Terraform with different Environments (Dev/Stage/Prod)**

- Introduction to Workspaces
- Use different values for different Workspaces
- Create a Development- and Production-Workspace
- Use of different Workspaces with Remote-States