

The DevOps Path: Mastering Infrastructure and Configuration as Code with Terraform and Ansible

Course 35658 – 40 Hours

Overview

Terraform has gained a lot of traction and popularity in the last two years, as more and more companies migrate to cloud or manage multiple Infrastructures (Hybrids). Terraform is the go to tool you need to master if you are or about to get into an Ops / DevOps role. While You typically use a technology like Ansible, Chef, or Puppet to automate the provisioning of software, Terraform starts from the same principle but focuses on the automation of the infrastructure itself. Your whole cloud infrastructure (instances, volumes, networking, IPs) can be described in Terraform. Use Terraform to describe your AWS, GCE, AZURE, OpenStack and many more in one place and in one code.

In this course you'll learn how to use Terraform to automate your infrastructure. we will teach our students the basics of Terraform and will then explain how to automate infrastructure on Cloud Infrastructure using Terraform and to the most advanced topics such as Templating, code reuse multi account and region deployments, ECS and EKS and many more.

Ansible Created a radical change in simplifying IT automation. Ansible Automation Engine became the most commonly used configuration management, application deployment and intra-service orchestration tool. Using Ansible you'll be able to automate every piece of the job in hand. In this course we'll master Ansible while tackling and learning base on real world use cases. We will starts with Ansible's architecture and how to provide effective scalable automation, then we move on to installing and configuring it with various platforms and into advance topics. We will learn how Ansible helps us to manage cloud services and containers and provision it with AWS and Docker. Finally, you will master troubleshooting, validating, and testing with Ansible.

The Difference Between Ansible and Terraform and why to learn both

Some will say, and we agree, that the two serve the same market segments, and overlapping in the use cases they solve. But really they overlap in the provisioning part, as stated in the table below. So how are they different? Where do you use one vs the other?

With Terraform you are **ONLY** building infrastructure (cloud, VMware, etc). You or your Team are always going to need further orchestration and configuration management leveraging tools - like Ansible.

Once Terraform has built the infrastructure, we will kickstart Ansible to finish off the heavy lifting to get the machines/containers configured and working. This can be a very cohesive workflow, and plays to the strengths of both tools.

On the more advanced front, Terraform requires you to be ready to treat infrastructure truly as code. Your entire plan/blueprint exists as a part of your Terraform plan, with which you then apply to your targeted endpoint (AWS, Azure, Google, VMware) to build machines.

High-level view of where each tool could play:

<u>Tool</u>	<u>Task</u>
Terraform	Provisioning / De-provisioning of machines
Ansible	Orchestration, possibly including orchestrating the Terraform build
Ansible	Configuration and advanced deployment of applications and services
Terraform	Network automation
Ansible	Security, patching, and compliance automation

On Completion, Delegates will be able to

- Use Terraform to automate your infrastructure
- Automate infrastructure on cloud providers using Terraform (including templating, code reuse multi account and region deployments for Containers, ECS, EKS, EC2)
- Master Ansible and its architecture
 - Provide effective scalable automation with Ansible
 - Install and configure Ansible with various platforms
- Manage cloud services and container with Ansible
- Provision Ansible with AWS and Docker
- Master troubleshooting, validating and testing with Ansible

Who Should Attend

- DevOps
- IT Personal
- CI/CD Professionals
- Operation Managers

Prerequisites

- Linux
- Familiarity with AWS (we will do our labs on AWS)

Course Contents

Part 1 – Terraform

Terraform Introduction

- Installing Terraform and Setting up AWS environment
- Introduction to Terraform

Terraform Basics

- Variables

- Software provisioning
- Outputs and module outputs
- State and Remote state using dynamoDB
- Data sources - Sharing state across modules
- Templates
- Modules
- Terraform Basics Quiz and Labs

Terraform with AWS

- AWS Infrastructure OverView
- Creating and managing -
 - VPC
 - ALB and Target Groups
- ASG
- UserData
- Static IP, EIP's and Route53 along with routing
- RDS Management
- IAM & IAM Roles Management
- ECS & EKS

Advance Terraform

- Interpolation
- Conditionals
- Built-in functions
- Project layout

PIPELINE DEMO

- Jenkins with Terraform and Packer

Part 2 – Ansible

Ansible Introduction

- Installing Ansible and setting up AWS environment
- Introduction to Ansible

Ansible Architecture and Design

- Ansible inventories
- Ansible Modules
- Yaml
- PlayBooks
- Variables
- Facts
- Templating Jinja2
- Creating and Executing

Ansible Advanced

- Modules
- Dynamic inventories
- Registers
- Looping
- Async and Parallel
- Tasks
- Magic variables
- Blocks
- Vault
- Custom Modules and Plugins
- Includes and Imports, Tags

Ansible with AWS and Containers

- Ansible and AWS
- Ansible and Docker

Part 3 – Terraform and Ansible - From Code to Production - Hands on final project

- Building Full AWS Production environment with Terraform base on ECS / EKS
- Building custom Docker images with Ansible and their configuration
 - Pushing to Docker Repo
 - Automated Testing of the created containers
- Automated Deploying of the new containers to Sandbox & Production
 - Terraform on the fly provisioning of a Sandbox environment
 - Deploying containers to Sandbox
 - Smoke Test
 - Destroying Sandbox
 - Terraform Provisioning of Production Environment with Green Blue Deploy

Technical Requirements:

An AWS account - free account to do the labs provided by the student / company.