Deploying Applications to the Cloud using Terraform

This class is designed to teach DevOps engineers, Production Support staff and developers how to effectively use Terraform to manage infrastructure in a cloud environment. The course covers the best practices around the concept of infrastructure as code and then introduces how this can be best be applied using Terraform.

# Course Duration

3 days

# Who it is for

Developers, Devops engineers, cloud architects

# Prerequisites

* Basic AWS Cloud Skills would be beneficial
* Basic understanding of deployment practices would be beneficial

# Objectives

At the end of this course, students should be able to:

# Course Contents

## Infrastructure as Code and Automated Deployments

* What is IaC
* Approaches to IaC
* Terraform, Cloudformation, SAM and others

## Infrastructure in Source Control – Git

* Introduction to Git
* Working with Git in the Enterprise
* Maintaining IaC in Git
* Triggering pipelines from Git hooks

## Effective Git Usage

* Working with a local Git repository
* Working with a remote git repository
* Effective use of branching
* Handling conflicts
* Git best practices and pull requests

## IaC and DevOps

* Performing automated deployments
* Blue / Green
* Canary
* In place rolling updates

## Terraform Fundamentals

* Understanding how Terraform Works
* Setting up Terraform
* Deploying a single server using Terraform
* Deploying a load balanced cluster of servers using Terraform

## Understanding the Terraform Language HCL

* Arguments and Blocks
* Identifiers
* Comments
* Encoding
* JSON syntax
* Style conventions

## Terraform Datatypes

* Working with strings and numbers
* Working with collection types
* Working with object types
* Using variables as input and output values

## Terraform Functions

* String and number functions
* Collection related functions
* Security related functions
* Network related functions
* File system and date related functions

## Terraform Constructs

* Looping with for\_each
* Working with counters
* The significance of depends\_on
* Working with different Providers
* Using Meta-arguments

## The Terraform CLI

* Understanding the core commands
* Validating using validate
* Using init, plan and apply

## Working with Terraform Modules

* Using registry based modules
* Creating custom modules
* Publishing custom modules to a registry

## Managing State using Terraform

* How do you know what you have
* Maintaining a registry of resources
* The Terraform enterprise solution
* Deploying, tracking and updating deployments using Terraform enterprise

## Configuration Management with Ansible

* Introduction to Configuration management tools
* Automating configuration using Ansible
* Configuring servers using Ansible
* Deploying an updated application using Ansible

## When to use Ansible or Terraform

* What should generally be done in Terraform
* What should generally done using Ansible
* Combining Terraform and Ansible

## Extended Project

* You are provided a sample Spring/React fullstack application
* Deploy this application to AWS using either containers or EC2
* Set up an appropriate pipeline for updates to be deployed