Terraform training plan-

What you'll learn

Everyone will learn from basics of terraform syntax to advanced terraform code writing for complex AWS services such as building Auto Scale Group of Cluster of Web Servers with a load balancer.

What is terraform state and how that can be used in Terraform stack of resources and can help large team

What is Terraform module and why it is powerful. Create Terraform module and discover module gotchas. How to create and access Terraform Module Inputs and Outputs.

How to work with key AWS services such as IAM , VPC ,Route53,EBS and more.

Create identical Infrastructure stack using Terraform workspace and tfvar.

Terraform key concepts such Interpolation , remote state , output attributes , variables , useful commands data resource with real examples.

Day 1

 What is Infrastructure as a Code

 Introduction to Terraform

o What is Terraform

o How Terraform works

o Understanding Cloud Orchestration code and Terraform

o Why Terraform

o Brief History of Terraform

o Terraform vs Other IaC Tools

 Terraform Installations

o Window

o Linux

o Setting up environmental variable

o Validate Installation by Terraform Version command

 Fundamental Concepts of Terraform

o Terraform Basic

o Version

o Terraform Plan, Show, Apply, Destroy

o Terraform registry

o Interpolation

 Terraform Programming Structure

Day 2

o Terraform Syntax: Providers

o Terraform Syntax: Resources

o Terraform Syntax: Variables

o Terraform Syntax: Data Sources

o Terraform Syntax: Outputs

o Terraform: Loops & If-Statements

o Terraform: Conditions

o Terraform: Build in Functions

 Provisioning Resources in Azure using Terraform

o Different ways of authenticating to Microsoft Azure using Azure Provider

o Understanding Resources in Terraform

o Create resources in Azure using Terraform

o Change Resources in Azure using Terraform

o Destroy Resources in Azure using Terraform

 Deployment Automation using Terraform

o Deploying a Vanilla Windows Server 2016 Azure VM with Terraform

o Deploying a Vanilla Linux Server (RHCE) Azure VM with Terraform

o Deploying a Windows Server 2016 Azure VM with Terraform with IIS enabled and

configured

o Deploying a RHCE Linux Azure VM with Terraform with NGINX Web Server enabled and configured

o Deploying Azure VM's for a three-tier sample application architecture through Terraform on IaaS

o Deploying AD DC with two-member server on Azure VM through Terraform

o Deploying a three-tier sample application architecture through Terraform, with Azure SQL DB as the data-tier

Day 3

 Terraform Modules

o Introduction to Modules

o Module repositories

o Refactoring the existing configuration into a Module

o Module Output

o Module versioning

 Terraform State Management

o Understanding Terraform State

o Shared Storage for State Files

o Isolating State files by Workspace /File Layout

 Terraform Troubleshooting and Testing

o Debugging the Terraform Script

o Manual Testing

o Automated Testing

 Terraform Integration and Extensions

o Integrating Terraform with Git

o Integrating Terraform with Packer

o Integrating terraform with other IaC Tools

o Terraform templates

o Terraform plugins

Day 4

Aws Terraform Course

Stop using root user and create separate admin user for day to day work.

Install and configure AWS Command line on your system

Install Terraform binary in your system

Install AWS Command line tool and Terraform binary on Windows

Code Editor : VSCode

Ec2

Clone this GITHUB repo

Deploy single server : EC2 Instance

Deploy single web server - using user_data

Deploy single configurable web server using variable

Deploy cluster of web servers in Auto Scaling Group

Deploy cluster of webs servers in Auto Scaling Group with ALB

Terraform Key Concept

Terraform Interpolation : most useful topic !

Terraform variables

Terraform outputs

Terraform remote state : why its so important and useful ?

Terraform data resource

Day 5

Terraform Module

Module basics

What is module ? Whats the syntax to declare module ? Every Terraform configuration has at least one module, known as its root module, which consists of the resources defined in the .tf files in the main working directory.

Module Inputs

what is module inputs ? How can you declare module inputs ? Module variable may or may not have default value assigned.

Module output

whats the syntax to defile module output ? how can you access them from other terraform stack ? Module is very power for this output feature as multiple module can work together for large infrastructure and it increases reusability.

Module gotchas

Two very important gotcha you should remember as you are writing more complex terraform code. Always use the interpolation "${path.module}" for file paths in module. And always prefer to use separate resource than inline resources , I will demonstrate you how to use them.

Module Versioning

How can you update your module when it is used by 100s of other team members? Your update may break their terraform stack who are using to create their infrastructure. Module versioning is the solution for that. You use git repository as source of your module. Then do a "git tag" to your update and use the tag information in terraform module source argument. Go through the demonstration and apply in your project.

Terraform State

what is remote state storage ? why we need ?

Terraform State locking

Isolate terraform state

Working with Key AWS Service

create AWS IAM roles and attach policy

create route53 records

create VPC , Private and Public subnet , Internet Gateway and NAT Gateway

Terraform Provisioner

provisioner local-exec

provisioner remote-exec

Terraform WOrkspace and Tfvars

workspace commands

workspace lab demonstration

Create identical infrastructure stack using .tfvar

Real World Terraform Project

Deploy an Jenkins Server

Project 2

Deploy a VPC

Remote State in Terraform Cloud

VPC Subnets "Hardcoded"

VPC Subnets - Terraform Apply Yourself!

The cidrsubnet() Function

The aws_availability_zones Data Source

The random_shuffle Resource

Why didn't we use for_each?

Route Tables and the Internet Gateway

VPC Security Groups "Hardcoded"

VPC Security Groups Dynamic

VPC Security Groups - Terraform Apply Yourself!

VPC RDS Subnet Group and Conditionals

Basic RDS Setup

VPC Outputs for RDS - Terraform Apply Yourself!

Managing Sensitive RDS Values

ALB Initial Setup

ALB Security Group and Subnets - Terraform Apply Yourself!

ALB Target Group and the uuid and substr functions

ALB Listener

Upgrading Terraform (0.14.1+ required for ignore_changes)

The aws_ami Data Source

Scaffolding Our EC2 Instance and random_id

EC2 - Terraform Apply Yourself!

SSH Key for Our Instance and the file Function

Controlling random_id Changes with Keepers

EC2 User Data and Template Files

EC2 User Data Template Variables - Terraform Apply Yourself!