TERRAFORM TRAINING

Level - Advance

COTOCUS

JUNE 1, 2022

COTOCUS LTD.
Bangalore, India

TRAINING DATES	TBD
TIME	TBD
TRAINER NAME	RAJESH KUMAR

AGENDA

DAY 1

- What is DevOps Vs DevSecOps Vs SRE? – Intro Level
- Tool sets in DevOps Vs DevSecOps Vs SRE? – Intro Level
- Overview of infrastructure during SDLC?
- Problems with many infrastructure tools?
- Introducing Infrastructure as Code
- Declarative vs. Imperative
- Introduction of Terraform
- Advantage & Disadvantage of Terraform
- How Terraform works for Infrastructure as Code?
- Alternative of Terraform
- Terraform Use cases
- How to install Terraform?
    - Introductions of Terraform Components?
    - - Terraform Providers
    - - Terraform Registry
    - - Terraform Resources
    - - Terraform Resources Argument Reference
- Deploying Your First Terraform Configuration
- Terraform Basics Workflow using Terraform CLI
    - terraform validate
    - terraform init
    - terraform plan
    - terraform apply
    - terraform show
    - terraform detroy
- Updating Your Configuration with More Resources
- Configuring a Resource After Creation
- Adding a New Provider to Your Configuration
- Undertanding Terraform state file
- Terraform Resource Behavior & Lifecycle

DAY 2

- Understanding Terraform HCL syntax & Style
    - Terraform Resources Meta-Argument

    - depends_on
    - count
    - for_each
    - provider
    - lifecycle
- Using Terraform for Create Cloud Provider Compute Resources
- Introduction of Terraform Variables & Values
- - Input Variables
- - Output Values
- - Local Values
- Where to declare a Terraform Input Variables?
- Deep dive into Types of Terraform Input Variables
    - - string
    - - number
    - - bool
    - - list
    - - set
    - - map
    - - object
    - - tuple
- How to access Terraform Input Variables value?
- Terraform Input Variables precedence & Scope
- Working with Terraform Output Values
- Working with Terraform Local Values
- Introduction of Terraform Data sources?
- Accessing Terraform Data sources?
- Working with Terraform HCL Operators
- Overview of Terraform Functions

DAY 3

- Working with Terraform Functions
    - - Numeric Functions
    - - String Functions
    - - Collection Functions
    - - Encoding Functions
    - - Filesystem Functions
    - - Date and Time Functions
    - - Hash and Crypto Functions

    - IP Network Functions
    - Type Conversion Functions
    - Introducing Terraform Provisioners
    - Working with Terraform Provisioners
    - file
    - local-exec
    - remote-exec
- Using Terraform for Create Cloud Provider Storage & Networking Resources
- Working with Terraform Templates
- Working with Workspaces for multiple environments
- Working with Remote Backend for managing State file for team
    - artifactory
    - s3
    - azurerm
    - gcs
- Understanding Terraform State locking
- Terraform Troubleshooting using logs & common errors
- Introducing Terraform Module
- Using with Terraform Module from Registry
- Developing Custom Terraform Module
- Publishing Modules on the Terraform Registry
- Introducing Terraform Console
- Introducing Terraform Tags
- Introducing Terraform Cloud
- Using Terraform for use Multiple providers for CI/CD

DAY 4

Azure RM Templates

- **Introduction to Azure ARM Templates**
- **Components of Azure ARM templates**
- **Review Azure ARM template in Azure portal**
- **Deep dive into Azure ARM templates**
- **Understanding Azure ARM template components**
- **Azure ARM template structure**
- **Azure ARM template - best practices**
- **Deep dive into Azure ARM templates**
- **Deep dive into Azure ARM templates #2**
- **Deploy Azure ARM template**
- **Nested Azure ARM templates**
- **Deploy nested Azure ARM templates**
- **Azure ARM template deployment modes**
- **Azure ARM template deployment modes #2**

DAY 5

- Collaborate on version-controlled configuration using Terraform Cloud. Follow this track to build, change, and destroy infrastructure using remote runs and state.
- TERRAFORM CLOUD: Sign up for Terraform Cloud
- Sign up for Terraform Cloud, which provides free remote state storage, a stable run environment, version control system (VCS).
- TERRAFORM CLOUD: Create a Workspace
- Create a Terraform Cloud workspace. Fork a demonstration GitHub repo containing Terraform configuration
- TERRAFORM CLOUD: Create Infrastructure
- Configure a Terraform Cloud workspace with Cloud credentials by setting environment variables. Set DynamoDB read and write.
- TERRAFORM CLOUD: Change Infrastructure
- Queue a speculative plan by opening a pull request. Inspect the plan and merge the PR to automatically queue a Terraform Cloud.
- TERRAFORM CLOUD: Destroy Resources and Workspaces
- Destroy the resources in a Terraform Cloud workspace, and delete the workspace via the web UI.
- TEAM & GOVERNANCE: Enforce a Policy
- Create a version-controlled policy to check the Terraform version using Sentinel, a policy-as-code platform. Fork a demo repo.
- TEAM & GOVERNANCE: Control Costs with Policies
- Turn on cost estimation in your Terraform Cloud organization. Write a soft-mandatory policy against example infrastructure to.
- TERRAFORM ENTERPRISE:
    - - Architecture
    - - Pre-Install Checklist
    - - Install and Configure
    - - Basic worlflow of Terraform Enterprise
    - - User Management
    - - Troubleshooting & Logging & Monitoring
    - - General & System Settings
    - - Backups and Restores
    - - Getting strated with Terraform Enterprise Dashboard
- Introducing Terraform Cloud
- Using Terraform for use Multiple providers for CI/CD

LAB TOPICS

Terraform

1.1_Creating Base configuration and First deployment

1.2_Setup an Azure provider

1.3_Learn Terraform workflow commands

1.4_Deploy our first terraform configuration file

2.1_Learn to work with more terraform commands

2.2_Modifying resources and review the changes

2.3_Destroy resources and review the changes

2.4_Import resources to state file and data sources.

3.1_Setting up Frontend

3.2_Setting up Backend

3.3_Setting up the Jump box environment

3.4_Reviewing the whole lab together

4.1_Output Variables

4.2_Remote State Storage

4.3_Interpolation of variables values

5.1_Module using local paths

5.2_Modules using terraform registry

6.1_Implicit Dependencies

6.2_Managing Multiple Environments Using Workspaces.

6.3_Managing Multiple Environments Using Directories

Update on Terraform-v14

Configuring Virtual Host using Managed Identity.

7.1_Azure Boards, Repos and Pipelines

7.2_ Azure Test Plans, Artifacts and DevOps Server

7.3_Git Version Control System

8.1_Secret Management using keyvault as datasource

8.2_Using key vault with service principal

8.3_Using secrets and Keyvault in devops

8.4_Using service connection in devops

9.1_Setting up release pipeline using workspace

9.2_Setting up release pipeline using directories

9.3_Continuous delivery, triggers and approvals

Step-02: Install Azure DevOps Terraform Plugins in Azure DevOps Organization

Step-03: Review Terraform Manifests, Kubernetes Manifests and Pipeline backup fi

Step-04: Setup Github repository local and remote and copy k8s and terraform man

Step-05: Create Service Connection, Fix AD Permissions, Create SSH Key and Uploa

Step-06: Create Pipeline with Terraform Validate Stage - Part-1

Step-07: Create Pipeline with Terraform Validate Stage - Part-2

Step-08: Introduction to Deploy Dev AKS Cluster Deployment Job in Stage 2 of Pip

Step-09: Write Pipeline code to Provision Dev AKS Cluster

Step-10: Verify Dev AKS Cluster Provisioning is successful using Azure DevOps Pi

Step-11: Create QA envionment related Pipeline code and Provision QA environment

Step-12: Verify QA Environment

Step-13: Add new nodepool, check-in code, monitor pipeline and verify changes

Lab - Azure Pipelines - Azure Terraform

Lab - Azure Pipelines - Azure Terraform – Resources