Faysal Aziz

Introduction to Terraform Foundations for AWS

2-day course hands on workshop

FA-0263 · 2-day

DevOps, Cloud & Infrastructure

Introduction

Terraform, an open-source IaC tool created by HashiCorp, is at the forefront of this revolution. It enables organizations to define, provision, and manage infrastructure across various cloud providers, including AWS, in a consistent and reproducible manner.

This course, "Terraform Foundations for AWS," is designed to provide you with the essential knowledge and skills to leverage Terraform for managing AWS resources. By the end of this course, you will be equipped to automate infrastructure deployment, enhance scalability, and improve the reliability of your cloud-based solutions.

Learning outcomes

By the end of this course, participants will be able to:

- Understand the fundamentals of Infrastructure as Code (IaC) and its benefits.
- Set up and configure Terraform for AWS.
- Write and manage Terraform configuration files.
- Utilize Terraform modules to create reusable and maintainable code.
- Implement best practices for managing Terraform state.
- Use GitLab for version control and collaboration on Terraform projects.
- Deploy and manage AWS resources using Terraform.
- Troubleshoot common issues and optimize Terraform configurations.

Prerequisites

- Good understanding of cloud computing concepts.
- Experience with AWS services and concepts.
- Working knowledge of Git and GitLab.
- Command-line interface (CLI) experience.
- A laptop with internet access and the ability to install software.
- Access VS Code, ability to SSH to remote lab server.

Training outline

1. Getting Started with Terraform

1. Installing Terraform on your local machine.
2. Configuring the AWS CLI and setting up credentials.
3. Writing your first Terraform configuration file.
4. Initializing a Terraform working directory.

2. Understanding Terraform Basics

1. Key Terraform concepts: providers, resources, variables, and outputs.
2. Writing and applying basic Terraform configurations.
3. Understanding the Terraform workflow: init, plan, apply, and destroy.
4. Managing Terraform state files and the significance of state in Terraform.

3. Terraform Configuration Files

1. Structure and syntax of Terraform configuration files.
2. Using variables for configuration flexibility.
3. Outputs and their role in Terraform.
4. Organizing configuration files for readability and maintainability.

4. Managing AWS Resources with Terraform

1. Creating and managing AWS EC2 instances.
2. Working with VPCs, subnets, and security groups.
3. Managing AWS S3 buckets and IAM roles/policies.

4. Provisioning RDS instances and other AWS services.

5. Terraform Modules

1. Introduction to Terraform modules and their benefits.
2. Creating and using local modules.
3. Exploring the Terraform Registry for reusable modules.
4. Best practices for writing and maintaining modules.

6. State Management

1. Understanding Terraform state and its importance.
2. Storing state files securely (e.g., using AWS S3 and DynamoDB).
3. Managing state file locking and state file integrity.
4. Terraform state commands and troubleshooting state-related issues.

7. Version Control with GitLab

1. Introduction to GitLab and its features.
2. Setting up a GitLab repository for Terraform projects.
3. Best practices for version control in Terraform projects.
4. Using GitLab CI/CD for automated Terraform deployments.

8. Best Practices and Troubleshooting

1. Terraform best practices for writing and managing configurations.
2. Common issues and troubleshooting tips.
3. Performance optimization techniques for Terraform.
4. Keeping your Terraform setup secure and compliant.

This comprehensive outline will guide you through the fundamentals of Terraform and equip you with practical skills to manage AWS infrastructure efficiently.

© 2026 Faysal Aziz · Kuala Lumpur