Terraform

1 week plan

Astragen.tech

Day 1: Introduction to Terraform

- Objective: Understand the basic concepts of Terraform and set up the environment.

- Topics:
  - Introduction to Infrastructure as Code (IaC)
  - Overview of Terraform: Purpose and use cases
  - Key Terraform concepts: Providers, Resources, Modules, and State
  - Installation of Terraform

- Practice:
  - Install Terraform on your local machine.
  - Write and run a simple Terraform configuration to create a basic resource (e.g., a local file or an AWS S3 bucket).

Day 2: Understanding Providers and Basic Configuration

- Objective: Learn about Terraform providers and how to define basic resources.

- Topics:
  - What are Terraform Providers?
  - Configuring providers in Terraform (e.g., AWS, Azure)
  - Writing and organizing Terraform configuration files (.tf files)

- Practice:
  - Configure an AWS provider.
  - Create an AWS EC2 instance using Terraform.
  - Explore Terraform commands: terraform init, terraform plan, terraform apply, and terraform destroy.

Day 3: Variables, Outputs, and State

- Objective: Understand the use of variables, outputs, and state management in Terraform.

- Topics:
  - Defining and using variables in Terraform (.tfvars files)
  - Outputs and how to use them
  - Terraform state: Purpose, backend configuration, and management

- Practice:
  - Create a Terraform configuration using variables for more flexibility.
  - Use outputs to extract and display resource information.
  - Explore the state file and practice moving state to a remote backend (e.g., AWS S3).

Day 4: Modules and Reusability

- Objective: Learn how to create and use Terraform modules to promote reusability.

- Topics:
  - Introduction to Terraform modules
  - Creating your own modules
  - Using public Terraform modules from the Terraform Registry
  - Organizing and structuring Terraform code for large projects

- Practice:
  - Create a simple module for provisioning an AWS EC2 instance.
  - Refactor previous Terraform configurations to use modules.
  - Use a public module (e.g., VPC module) to create networking infrastructure.

Day 5: Advanced Terraform Features

- Objective: Explore advanced Terraform features like data sources, conditionals, and loops.

- Topics:
  - Data sources and how to use them in Terraform
  - Conditional expressions in Terraform
  - Using loops (for_each, count) to create multiple resources.

- Practice:
  - Use data sources to fetch existing AWS resources (e.g., AMI IDs).
  - Implement conditional resource creation.
  - Use loops to create multiple resources (e.g., multiple EC2 instances or security groups).

Day 6: Terraform Provisioners and Integrations

- Objective: Understand Terraform provisioners and how to integrate Terraform with other tools.

- Topics:
  - What are Terraform provisioners, and when to use them?
  - Types of provisioners (local-exec, remote-exec)
  - Integrating Terraform with CI/CD pipelines (e.g., Jenkins)
  - Terraform Cloud and Terraform Enterprise overview

- Practice:
  - Implement a local-exec provisioner to run a local script after resource creation.
  - Set up a basic CI/CD pipeline that applies Terraform configurations automatically.
  - Explore Terraform Cloud by creating a simple workspace.

Day 7: Review and Project

- Objective: Consolidate learning with a review and a hands-on project.

- Topics:
  - Review all key concepts learned throughout the week.
  - Discuss best practices in Terraform (e.g., version control, state management, security).

- Practice:
  - Work on a mini-project: Set up a complete infrastructure on AWS, including networking (VPC), compute (EC2), and storage (S3).
  - Review the project and identify areas for improvement or further exploration.