1. Introduction to Terraform

- **What is Terraform?**
    - Overview of Infrastructure as Code (IaC)
    - Benefits of using Terraform for managing cloud infrastructure
    - Understanding the Terraform workflow (Write, Plan, Apply, Destroy)
- **Terraform vs Other IaC Tools**
    - Terraform vs CloudFormation
    - Terraform vs Ansible
    - Terraform vs Puppet and Chef
- **Installing Terraform**
    - Installing Terraform on various platforms (Linux, macOS, Windows)
    - Verifying installation (`terraform version`)
    - Setting up Terraform CLI and environment

2. Terraform Basics

- **Terraform Configuration Files**
    - Overview of Terraform configuration files (`.tf` files)
    - Understanding the basic structure of a Terraform configuration (Provider, Resources, Outputs)
    - Basic configuration syntax and language
    - Using HCL (HashiCorp Configuration Language)
- **Terraform Providers**
    - Introduction to Providers in Terraform
    - Understanding Provider configuration (e.g., AWS, Azure, Google Cloud)

    - Setting up and configuring a Provider (e.g., AWS Access Keys)
- **Resources in Terraform**
    - Defining resources (e.g., EC2, S3, VPC, subnets)
    - Creating and managing resources with Terraform
    - Modifying existing resources using Terraform (`terraform apply`)

---

3. Terraform Variables and Outputs

- **Using Variables**
    - Declaring variables in Terraform (`variable` block)
    - Types of variables (strings, integers, lists, maps)
    - Default values and variable validation
    - Passing variable values (`var`, `var-file`)
- **Output Values**
    - Defining and using output variables (`output` block)
    - Outputting resource attributes (e.g., public IP, instance ID)
    - Using outputs to pass data between modules and configurations

---

4. Terraform State and Remote Backends

- **Understanding Terraform State**
    - What is Terraform state and why is it important?
    - The role of the `.tfstate` file
    - Local state vs remote state
    - Managing and securing Terraform state files
- **Working with Remote Backends**
    - Setting up remote backends (e.g., AWS S3, Azure Blob Storage)
    - Configuring remote state with versioning and locking

    - Using backend configuration for state storage
    - Benefits of using remote backends (team collaboration, state consistency)

---

5. Terraform Modules

- **Introduction to Modules**
    - What are modules in Terraform?
    - Creating and organizing reusable Terraform modules
    - Using modules from the Terraform Registry
    - Importing and using local and remote modules
- **Module Inputs and Outputs**
    - Defining and passing variables to modules
    - Output values from modules
    - Best practices for organizing and structuring modules
- **Terraform Module Structure**
    - Folder structure for a Terraform module
    - Best practices for writing reusable and maintainable modules

---

6. Terraform Provisioners and Taints

- **Provisioners**
    - What are provisioners and how are they used?
    - Types of provisioners: `local-exec`, `remote-exec`, `file`
    - When to use provisioners vs when to avoid them
    - Example: Using `remote-exec` for configuring an EC2 instance
- **Tainting Resources**
    - What is tainting in Terraform?
    - Manually tainting resources using `terraform taint`

- Understanding the impact of tainting and how it affects resource lifecycle

7. Terraform Cloud and Workspaces

- **Introduction to Terraform Cloud**
  - What is Terraform Cloud and how it differs from local execution?
  - Setting up a Terraform Cloud account and organization
  - Using Terraform Cloud for remote runs, collaboration, and team workflows
- **Terraform Workspaces**
  - What are workspaces in Terraform?
  - Creating and using workspaces for managing different environments (development, staging, production)
  - Switching between workspaces and managing state files across environments

8. Advanced Terraform Features

- **Terraform Graphs**
  - Visualizing infrastructure relationships with `terraform graph`
  - Understanding resource dependencies in large infrastructures
- **Working with Data Sources**
  - Using Terraform data sources to fetch data from external systems
  - Example: Fetching existing resources from AWS (e.g., VPCs, AMIs)
- **Terraform CLI and Automation**
  - Automating Terraform runs using CI/CD tools (e.g., Jenkins
  - Integrating Terraform into automated workflows for provisioning
  - Using Terraform with pipelines for continuous deployment and infrastructure management

Advanced Topics

- **Terraform Enterprise**
    - Introduction to Terraform Enterprise and its benefits over Terraform Cloud
    - Managing Terraform workspaces, policies, and teams in Terraform Enterprise
- **Integrating Terraform with Service Meshes**
    - Managing service mesh infrastructure (e.g., Istio) with Terraform
    - Using Terraform for service mesh configurations and deployments