Terraform – 100+ Lab Exercises (Basic, Intermediate, Advanced)

□ Basic Level (30+ Exercises)

Objective: Grasp Terraform fundamentals, syntax, and build your first infrastructure modules.

□ Terraform Fundamentals

- Install Terraform and set up CLI.
- Initialize a Terraform project.
- Understand `.tf` file structure and HCL syntax.
- Configure `main.tf`, `variables.tf`, and `outputs.tf`.
- Use `terraform init`, `plan`, `apply`, `destroy`.

☁□ Basic Resource Provisioning

- Launch an AWS EC2 instance.
- Create an S3 bucket with basic properties.
- Deploy an Azure Virtual Machine.
- Configure a GCP Compute Engine instance.
- Use AWS Provider block with credentials.

□ Variables and Outputs

- Use input variables with `default` values.
- Assign variables via `terraform.tfvars`.
- Output instance public IP and ID.
- Create local values and references.
- Understand interpolation syntax `${}`.

□ Basic Modules and State

- Write reusable Terraform modules.
- Call a local module from main config.
- Understand remote and local state files.
- Use `terraform output` to retrieve values.

- Enable basic state locking with S3 + DynamoDB (AWS).

□ Intermediate Level (40+ Exercises)

Objective: Automate infrastructure workflows using advanced modules, remote backends, and collaboration patterns.

□ Intermediate Resource Management

- Launch multi-tier AWS architecture (VPC, EC2, RDS).
- Create autoscaling groups and launch templates.
- Set up Azure Storage Account with replication.
- Use GCP VPC and firewall rules.
- Deploy Lambda functions via Terraform.

□ Modules, Data Sources & Outputs

- Create nested modules (network, compute, storage).
- Reference data sources (AMI IDs, existing VPCs).
- Use `for_each` and `count` for multiple resources.
- Implement lifecycle policies (create_before_destroy).
- Map outputs across environments.

□ Remote Backends & Workspaces

- Store Terraform state in AWS S3.
- Lock state with DynamoDB.
- Use Terraform Cloud remote backend.
- Manage environments using Workspaces (dev, prod).
- Use backends in enterprise Terraform.

□ Secrets & Security

- Store secrets in AWS Secrets Manager.
- Inject environment variables securely.
- Integrate with HashiCorp Vault.
- Use `sensitive = true` for variables.
- Encrypt backend state files.

□ Advanced Level (40+ Exercises)

Objective: Build production-grade, secure, and scalable infrastructure using modular design patterns and CI/CD automation.

□ Infrastructure Architecture

- Build multi-region architectures.
- Configure global load balancing.
- Design and deploy private/public subnet combinations.
- Integrate Terraform with Kubernetes (EKS, AKS, GKE).
- Implement scalable microservices deployment using Terraform + Helm.

□ Advanced Modules & Abstractions

- Publish modules to Terraform Registry.
- Use `depends_on` effectively across modules.
- Refactor monolithic configs into micro-modules.
- Validate module input/output interfaces.
- Use local-exec and remote-exec provisioners.

□ Testing & Validation

- Use `terraform validate` and `terraform fmt`.
- Write unit tests with **Terratest** (Go).
- Use `checkov`, `tflint`, and `tfsec` for security checks.
- Automate code scanning in GitHub Actions.
- Implement policy-as-code (OPA, Sentinel).

□ CI/CD Integration

- Integrate Terraform in GitHub Actions / GitLab CI.
- Automate plan-apply workflow with approvals.
- Use `terraform plan -out=tfplan` for pipelines.
- Generate reports and diffs for infrastructure changes.
- Trigger downstream jobs on plan completion.

□ Capstone Projects

- Build end-to-end CI/CD pipeline with Terraform on AWS.

- Design a DR (Disaster Recovery) ready architecture using Terraform.
- Migrate existing infrastructure to Terraform-managed code.
- Create a modular multi-cloud provisioning system.
- Build a serverless infrastructure using Lambda, API Gateway, and Terraform.

□ Tools & Technologies

- **Cloud Providers:** AWS, Azure, GCP
- **Tools:** Terraform Cloud, Terragrunt, Atlantis, Vault
- **Testing & Security:** Checkov, Terratest, Tfsec, TFLint
- **CI/CD:** GitHub Actions, GitLab CI, Jenkins, CircleCI