AWS EC2,S3,VPC using Terraform

Infrastructure as Code (IaC) - Lab Practical

# 1. Why Terraform? (IaC Intro)

- Terraform is an open-source Infrastructure as Code (IaC) tool.  
- Automates the provisioning of cloud resources (like EC2, S3, VPC).  
- Enables version-controlled, repeatable, and sharable infrastructure.  
- Cloud-agnostic: works with AWS, Azure, GCP, etc.  
- Uses declarative language called HCL (HashiCorp Configuration Language).

# INSTALLING TERRAFORM:

sudo yum install -y yum-utils shadow-utils

sudo yum-config-manager --add-repo https://rpm.releases.hashicorp.com/AmazonLinux/hashicorp.repo

sudo yum -y install terraform

configure role with full access.

# Configuration files:

it will have resource configuration.

here we write inputs for our resource

based on that input terraform will create the real world resources.

extension is .tf

# 2. Terraform Script to Launch EC2 Instance

Provider “aws” {

Region=”us-east-1”

}

resource "aws\_instance" "my\_ec2" {  
 ami = "ami-0c02fb55956c7d316" # Amazon Linux 2 AMI in us-east-1  
 instance\_type = "t2.micro”  
}

# 3. Terraform Script to Create S3 Bucket

resource "aws\_s3\_bucket" "my\_bucket" {  
 bucket = "my-unique-bucket-name-123456"  
 acl = "private"  
 tags = {  
 Name = "MyS3Bucket"  
 }  
}

# 4. Terraform Script to Create VPC

resource "aws\_vpc" "my\_vpc" {  
 cidr\_block = "10.0.0.0/16"  
 tags = {  
 Name = "MyVPC"  
 }  
}  
  
resource "aws\_subnet" "my\_subnet" {  
 vpc\_id = aws\_vpc.my\_vpc.id  
 cidr\_block = "10.0.1.0/24"  
 availability\_zone = "us-east-1a"  
 tags = {  
 Name = "MySubnet"  
 }  
}  
  
resource "aws\_internet\_gateway" "my\_igw" {  
 vpc\_id = aws\_vpc.my\_vpc.id  
 tags = {  
 Name = "MyIGW"  
 }  
}

# 5. How to Run Terraform (Step-by-Step)

1. Install Terraform on your machine  
2. Configure AWS CLI using `aws configure`  
3. Save the above code into a file named `main.tf`  
4. Open terminal in the folder and run:  
 - terraform init  
 - terraform plan  
 - terraform apply  
5. Type 'yes' when prompted to proceed  
6. Check AWS Console to verify the resources

# 6. Expected Output / Verification

- A VPC with CIDR block 10.0.0.0/16  
- A Subnet within the VPC (10.0.1.0/24)  
- An Internet Gateway attached to the VPC  
- A t2.micro EC2 instance running (check Public IP)  
- An S3 bucket with a unique name  
- Resources should have proper tags like Name = MyEC2