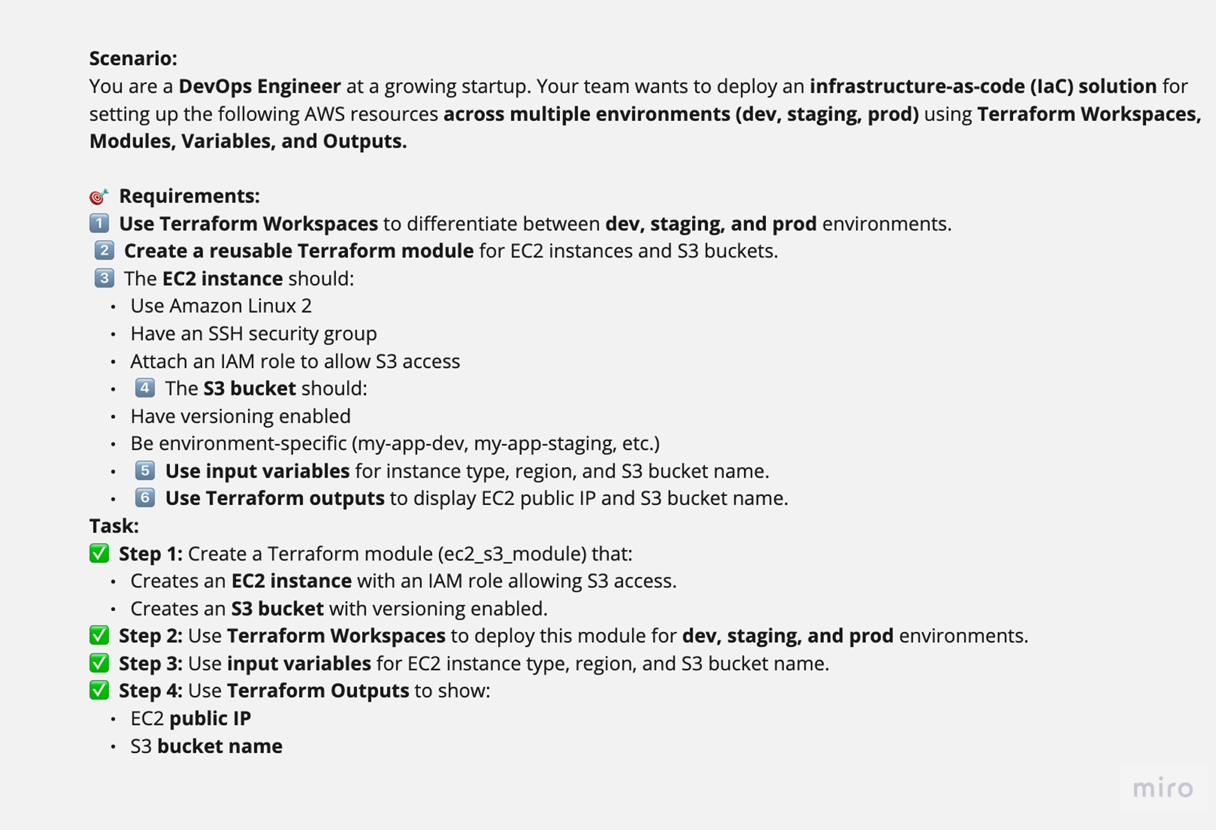
**Terraform Task**



Explanation :

1. Create the folder named terraform. Create a subfolder modules and create main.tf file in it.

**Main.tf file**

terraform {

  required\_providers {

    aws = {

      source  = "hashicorp/aws"

      version = "~> 5.0"

    }

  }

}

# Configure the AWS Provider

provider "aws" {

  region = var.aws\_region

}

#ssh security group

resource "aws\_security\_group" "allow\_ssh" {

  name        = "allow\_ssh-${terraform.workspace}"

  description = "Allow ssh access"

}

resource "aws\_vpc\_security\_group\_ingress\_rule" "ssh" {

  security\_group\_id = aws\_security\_group.allow\_ssh.id

  cidr\_ipv4         = "0.0.0.0/0"

  from\_port         = 22

  ip\_protocol       = "tcp"

  to\_port           = 22

}

resource "aws\_vpc\_security\_group\_egress\_rule" "allow\_all\_traffic\_ipv4" {

  security\_group\_id = aws\_security\_group.allow\_ssh.id

  cidr\_ipv4         = "0.0.0.0/0"

  ip\_protocol       = "-1" # semantically equivalent to all ports

}

# IAM role-aws\_iam\_role

resource "aws\_iam\_role" "s3\_access" {

  name = "s3\_access-${terraform.workspace}"

  # Terraform's "jsonencode" function converts a

  # Terraform expression result to valid JSON syntax.

  assume\_role\_policy = jsonencode({

    Version = "2012-10-17"

    Statement = [

      {

        Action = "sts:AssumeRole"

        Effect = "Allow"

        Principal = {

          Service = "ec2.amazonaws.com"

        }

      },

    ]

  })

}

# IAM role-aws\_iam\_role\_policy\_attachment

resource "aws\_iam\_role\_policy\_attachment" "s3\_policy" {

  role       = aws\_iam\_role.s3\_access.name

  policy\_arn = "arn:aws:iam::aws:policy/AmazonS3FullAccess"

}

# IAM role-aws\_iam\_instance\_profile

resource "aws\_iam\_instance\_profile" "s3\_profile" {

  name = "s3-profile-${terraform.workspace}"

  role = aws\_iam\_role.s3\_access.name

}

#ec2 instance

resource "aws\_instance" "ec2-instance" {

  ami                     = "ami-0c2e61fdcb5495691"

  instance\_type           = var.instance\_type

  key\_name = var.ssh\_key\_name

  vpc\_security\_group\_ids = [aws\_security\_group.allow\_ssh.id]

  associate\_public\_ip\_address = true

  iam\_instance\_profile   = aws\_iam\_instance\_profile.s3\_profile.name

}

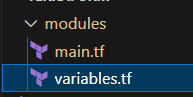
#s3 bucket

resource "aws\_s3\_bucket" "my\_s3" {

  bucket = "${var.s3\_bucket\_name}-${terraform.workspace}"

}

1. Create variable.tf under modules



**Variable.tf file**

variable "aws\_region" {

  description = "AWS region"

  type        = string

}

variable "instance\_type" {

  description = "Instance type"

  type        = string

}

variable "ssh\_key\_name" {

  description = "SSH key name"

  type        = string

}

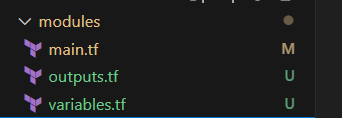
variable "s3\_bucket\_name" {

  description = "S3 bucket name"

  type        = string

}

1. **Create outputs.tf**

****

output "ec2\_public\_ip" {

value = aws\_instance.ec2-instance.public\_ip

}

output "s3\_bucket\_name" {

value = aws\_s3\_bucket.my\_s3.id

}

1. Create main.tf in the root directory

**main.tf – root**

provider "aws" {

  region = var.aws\_region

}

module "ec2-s3" {

  source          = "./modules"

  aws\_region      = var.aws\_region

  instance\_type   = var.instance\_type

  ssh\_key\_name    = var.ssh\_key\_name

  s3\_bucket\_name  = var.s3\_bucket\_name

}

output "ec2\_public\_ip" {

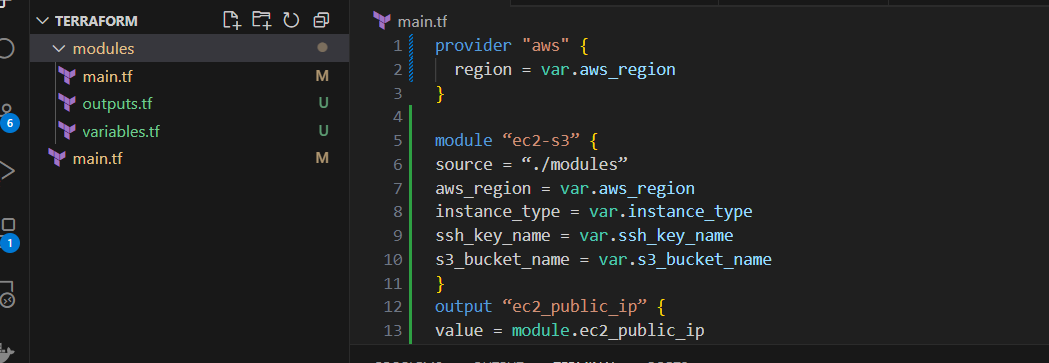
  value = module.ec2-s3.ec2\_public\_ip

}

output "s3\_bucket\_name" {

  value = module.ec2-s3.s3\_bucket\_name

}



1. Create terra.tfvars in root directory

**Terrs.tfvars**

aws\_region = "us-east-1"

instance\_type = "t2.micro"

ssh\_key\_name = "Nivetha"

s3\_bucket\_name = "nivi-bucket1"

create variable.tf in root directory

variable "aws\_region" {

description = "AWS region"

type = string

}

variable "instance\_type" {

description = "Instance type"

type = string

}

variable "ssh\_key\_name" {

description = "SSH key name"

type = string

}

variable "s3\_bucket\_name" {

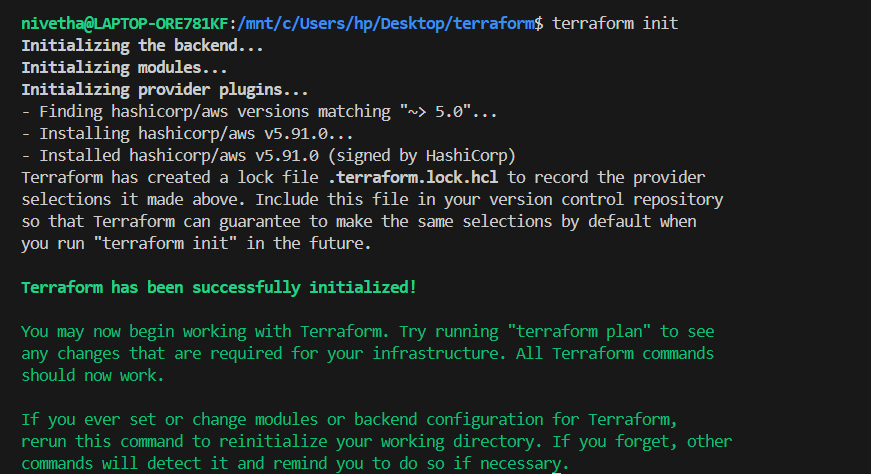
description = "S3 bucket name"

type = string

}

1. Create workspaces using the bellow commands

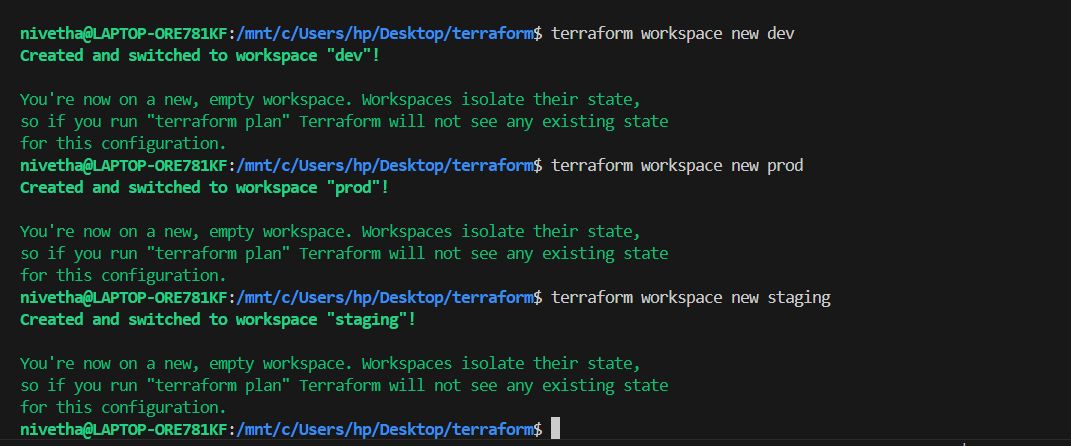
Terraform init



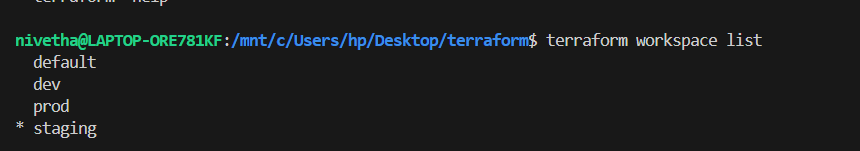
terraform workspace new dev

terraform workspace new prod

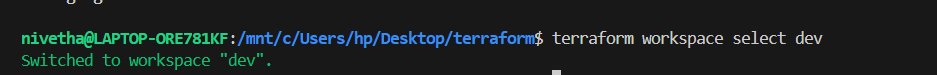
terraform workspace new staging



Terraform workspace list



terraform workspace select dev



terraform apply

