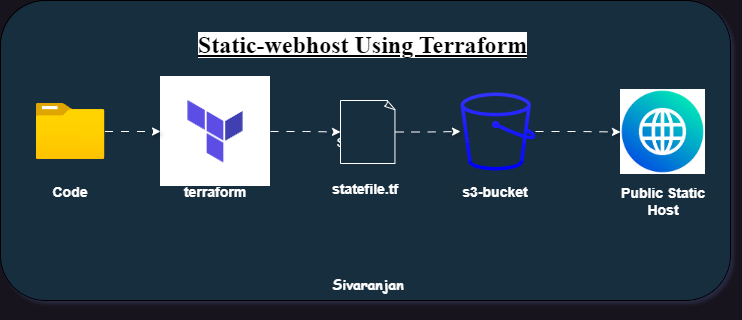
**S3 – Static\_WebHost Using Terraform**

****

**Step 1: Create Bucket**

resource "aws\_s3\_bucket" "mybucket" {

  bucket = "my-new-test-bucket-22"

}

**Step 2: Change Object Ownership**

#Object Ownership

resource "aws\_s3\_bucket\_ownership\_controls" "example" {

  bucket = aws\_s3\_bucket.mybucket.id

  rule {

    object\_ownership = "BucketOwnerPreferred"

  }

}

**Step 3: Make bucket Public Access**

#Bucket Public Access

resource "aws\_s3\_bucket\_public\_access\_block" "example" {

  bucket = aws\_s3\_bucket.mybucket.id

  block\_public\_acls       = false

  block\_public\_policy     = false

  ignore\_public\_acls      = false

  restrict\_public\_buckets = false

}

**Step 4: Bucket ACL**

#Bucket Acl

resource "aws\_s3\_bucket\_acl" "example" {

  depends\_on = [

    aws\_s3\_bucket\_ownership\_controls.example,

    aws\_s3\_bucket\_public\_access\_block.example,

  ]

  bucket = aws\_s3\_bucket.mybucket.id

  acl    = "public-read"

}

**Step 5: Object Upload**

#upload Code files

resource "null\_resource" "css" {

  provisioner "local-exec" {

    command = "aws s3 sync c:\\terraform-project\\s3\_static\_web\_host\\html  s3://my-new-test-bucket-22 --acl public-read"

    #acl    = "public-read"

  }

}

**Step 6: Static-web host configuration**

#website config

resource "aws\_s3\_bucket\_website\_configuration" "example" {

  bucket = aws\_s3\_bucket.mybucket.id

  index\_document {

    suffix = "index.html"

  }

  depends\_on = [aws\_s3\_bucket\_acl.example]

}

**Main.tf**

resource "aws\_s3\_bucket" "mybucket" {

  bucket = "my-new-test-bucket-22"

}

#Object Ownership

resource "aws\_s3\_bucket\_ownership\_controls" "example" {

  bucket = aws\_s3\_bucket.mybucket.id

  rule {

    object\_ownership = "BucketOwnerPreferred"

  }

}

#Bucket Public Access

resource "aws\_s3\_bucket\_public\_access\_block" "example" {

  bucket = aws\_s3\_bucket.mybucket.id

  block\_public\_acls       = false

  block\_public\_policy     = false

  ignore\_public\_acls      = false

  restrict\_public\_buckets = false

}

#Bucket Acl

resource "aws\_s3\_bucket\_acl" "example" {

  depends\_on = [

    aws\_s3\_bucket\_ownership\_controls.example,

    aws\_s3\_bucket\_public\_access\_block.example,

  ]

  bucket = aws\_s3\_bucket.mybucket.id

  acl    = "public-read"

}

#resource "aws\_s3\_object" "index" {

#  bucket = aws\_s3\_bucket.mybucket.id

#  key    = "index.html"

#  source = "index.html"

#  acl    = "public-read"

#  content\_type = "text/html"

#}

#upload Code files

resource "null\_resource" "css" {

  provisioner "local-exec" {

    command = "aws s3 sync c:\\terraform-project\\s3\_static\_web\_host\\html  s3://my-new-test-bucket-22 --acl public-read"

    #acl    = "public-read"

  }

}

#website config

resource "aws\_s3\_bucket\_website\_configuration" "example" {

  bucket = aws\_s3\_bucket.mybucket.id

  index\_document {

    suffix = "index.html"

  }

  depends\_on = [aws\_s3\_bucket\_acl.example]

}

**Step7: Providers.tf**

terraform {

  required\_providers {

    aws = {

      source  = "hashicorp/aws"

      version = "~> 4.0"

    }

  }

}

# Configure the AWS Provider

provider "aws" {

  region = var.aws\_region

}

**Step 8: Variables.tf**

variable "aws\_region"{

    description     = "Region in mumbai"

    type            =  string

    default         = "ap-south-1"

}

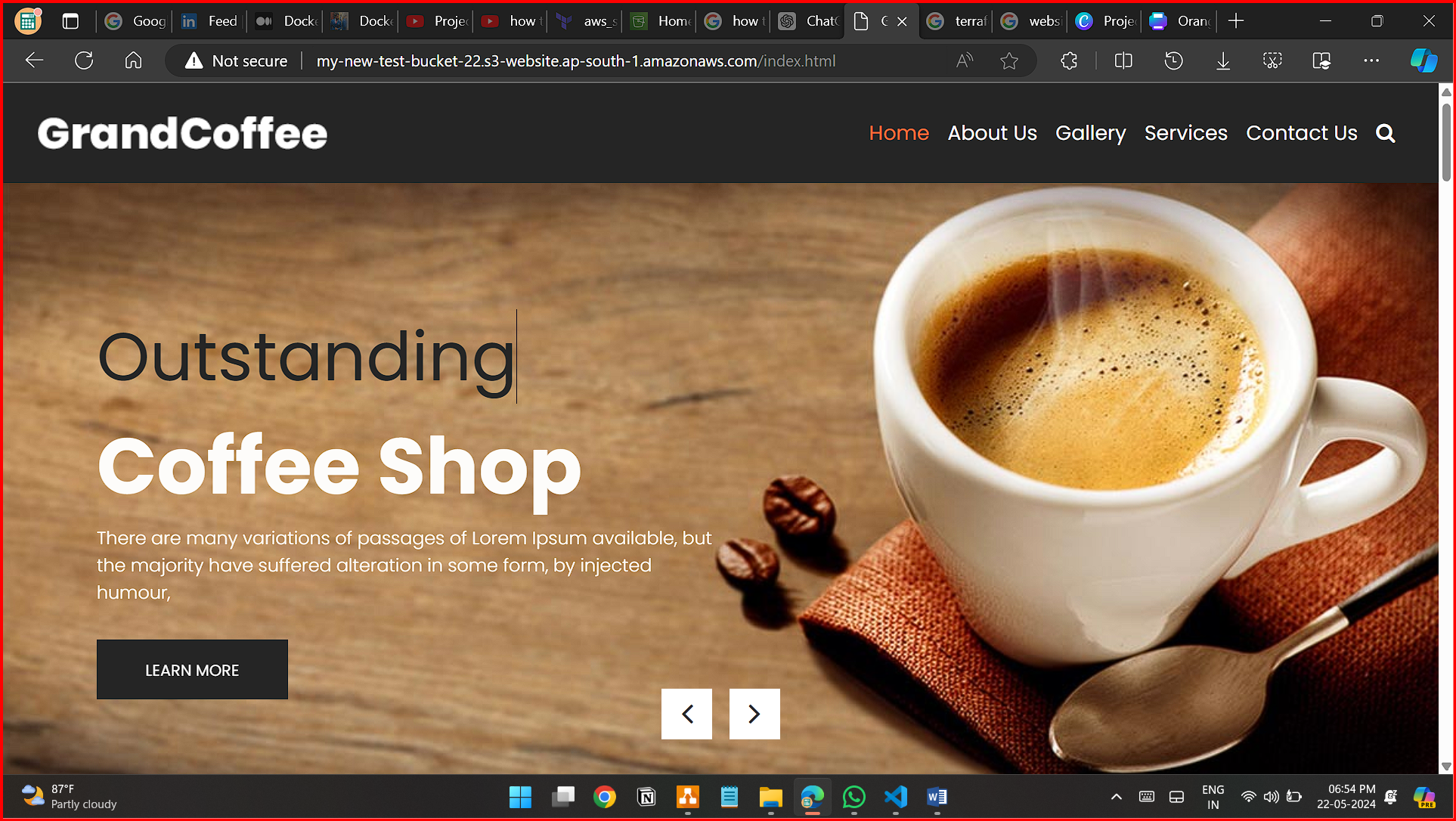
**Step 9: Outputs.tf**

output "websiteendpoint"{

    value = aws\_s3\_bucket.mybucket.website\_endpoint

}

**OUTPUT**

****