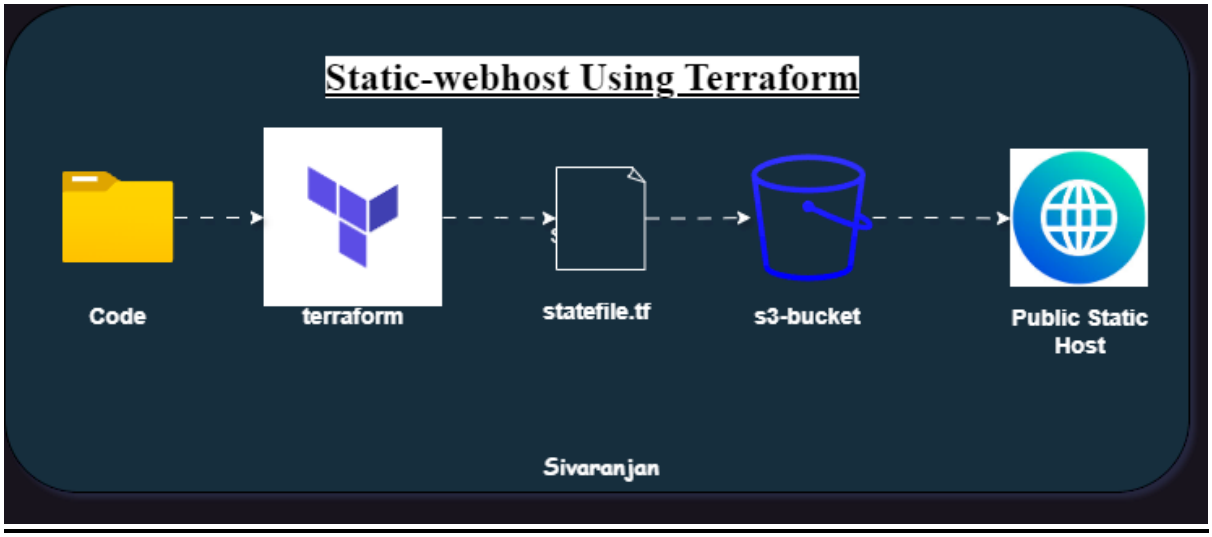
PROJECT

22-05-2024

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