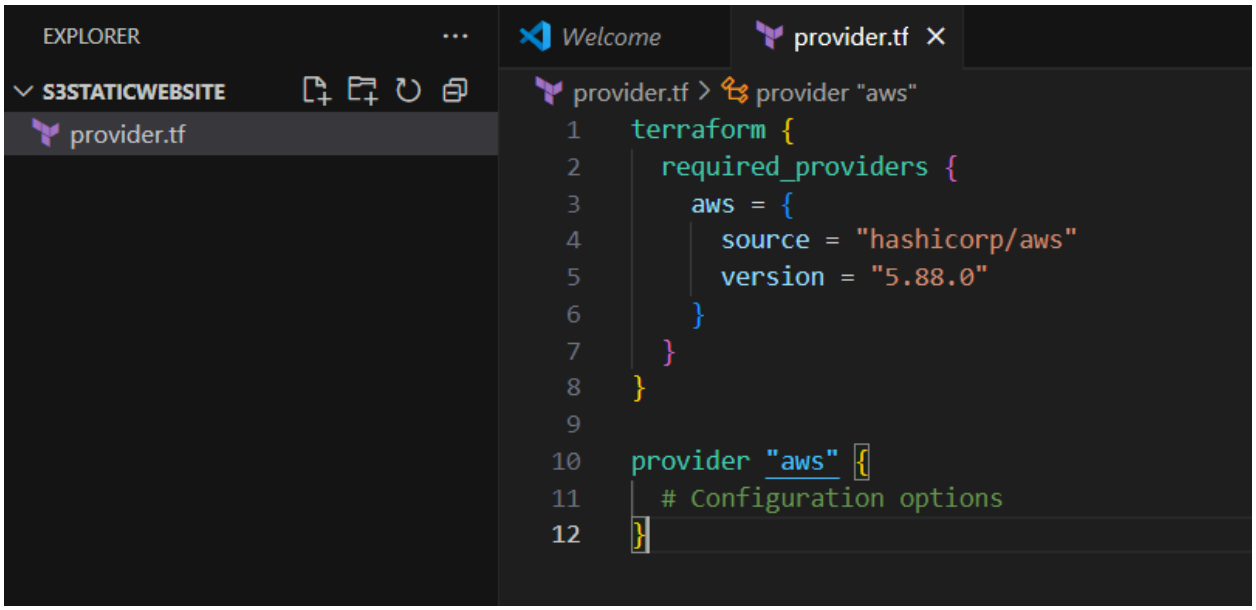
Simple Website Hosting using AWS S3 and Terraform

For this project, I will be using Terraform to create the resources I need to host a static website on S3.

```
C:\Users\Miguel Huerto\Desktop\terraform_course>cd ..

C:\Users\Miguel Huerto\Desktop>mkdir s3staticwebsite

C:\Users\Miguel Huerto\Desktop>cd s3staticwebsite

C:\Users\Miguel Huerto\Desktop\s3staticwebsite> code .
```




```
PS C:\Users\Miguel Huerto\Desktop\s3staticwebsite> terraform init
Initializing the backend...
Initializing provider plugins...
- Finding hashicorp/aws versions matching "5.88.0"...
- Installing hashicorp/aws v5.88.0...
```

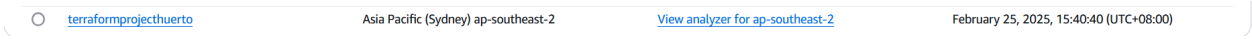
1. Create an S3 Bucket

main.tf > resource "aws_s3_bucket" "mybucket"

```
1  resource "aws_s3_bucket" "mybucket" {
2    bucket = var.bucketname
3  }
```

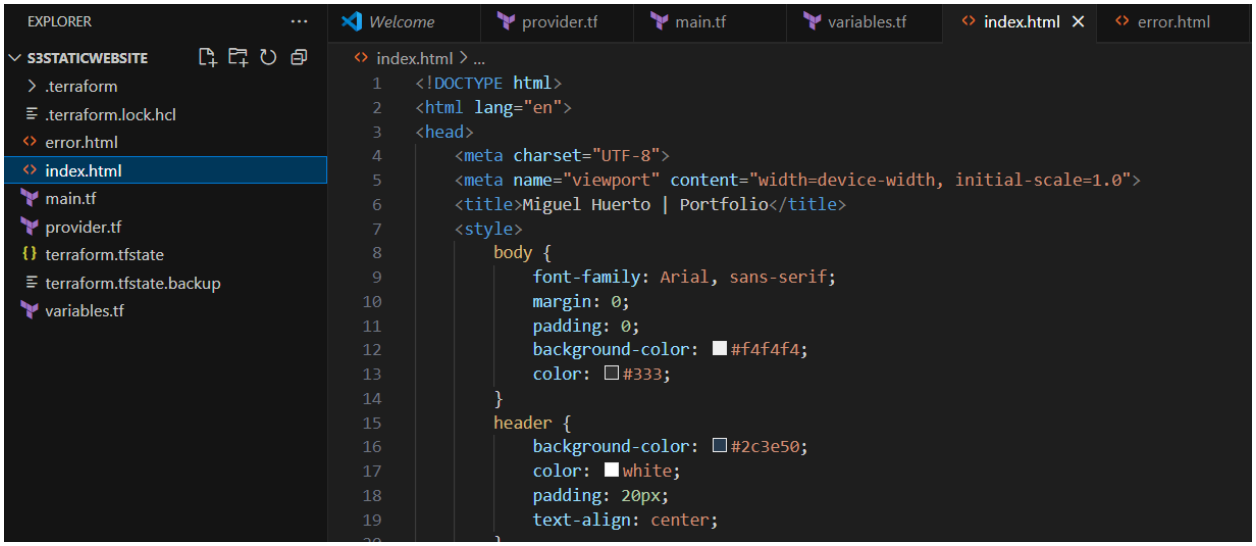
variables.tf > variable "bucketname"

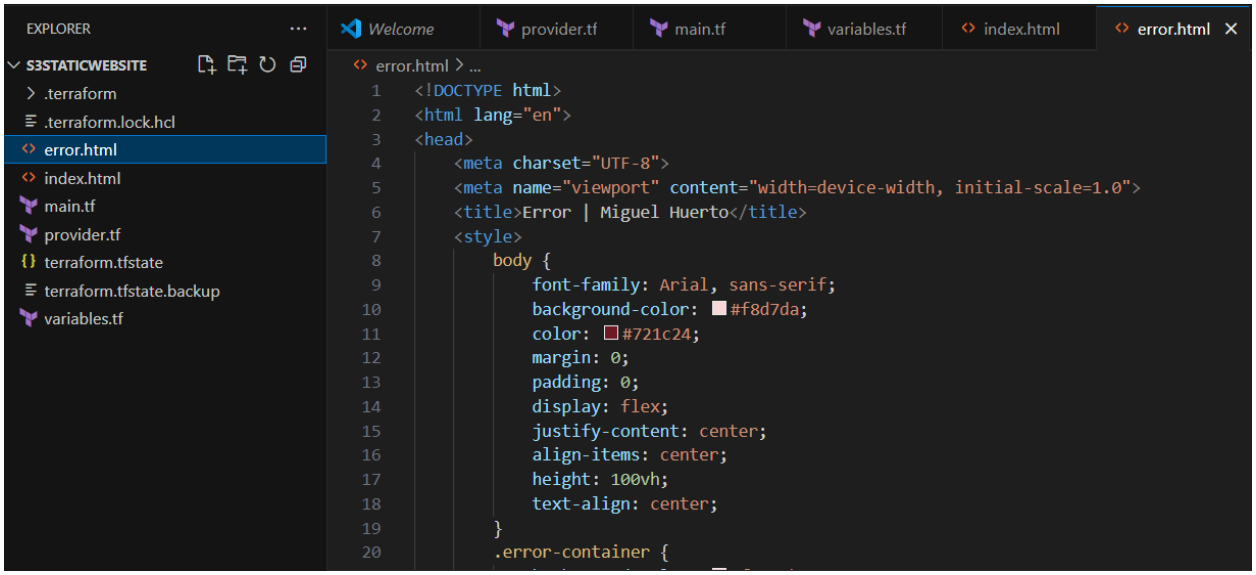
```
1  variable "bucketname" {
2    default = "terraformprojecthuerto"
3  }
```



2. Configure the bucket for Static Website Hosting

```
resource "aws_s3_bucket_ownership_controls" "example" {
  bucket = aws_s3_bucket.mybucket.id

  rule {
    object_ownership = "BucketOwnerPreferred"
  }
}

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
}

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"
}
```





3. Upload Website Files

```
resource "aws_s3_object" "index" {
  bucket = aws_s3_bucket.mybucket.id
  key    = "index.html"
  source = "index.html"
  acl = "public-read"
  content_type = "text/html"
}

resource "aws_s3_object" "error" {
  bucket = aws_s3_bucket.mybucket.id
  key    = "error.html"
  source = "error.html"
  acl = "public-read"
  content_type = "text/html"
}
```

```
resource "aws_s3_bucket_website_configuration" "example" {
  bucket = aws_s3_bucket.mybucket.id

  index_document {
    suffix = "index.html"
  }

  error_document {
    key = "error.html"
  }

  depends_on = [aws_s3_bucket_acl.example]
}
```

terraformprojecthuerto Info

Objects | Properties | Permissions | Metrics | Management | Access Points

Objects (2)

Copy S3 URI | Copy URL | Download | Open | Delete | Actions | Create folder | Upload

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

Find objects by prefix

1

Name	Type	Last modified	Size	Storage class
error.html	html	February 25, 2025, 16:20:46 (UTC+08:00)	1.3 KB	Standard
index.html	html	February 25, 2025, 16:20:46 (UTC+08:00)	3.0 KB	Standard

Miguel Huerto

Cloud Engineer | Tech Enthusiast

Projects | Education | Experience

Projects

Project One

Description of the first project, including technologies used and outcomes.

Project Two

Description of the second project, highlighting key features and user impact.

Educational Background

Bachelor of Science in Industrial Engineering

University of the Philippines Diliman | 2022 | Cum Laude

Certifications

Relevant certifications and completion dates.