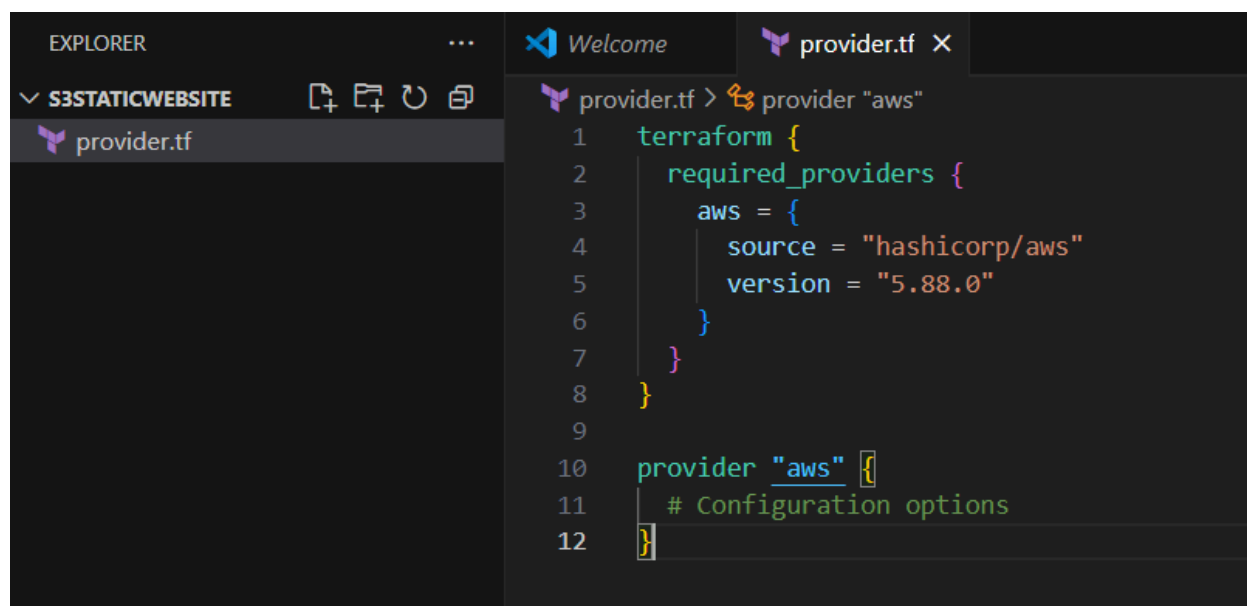


# Simple Website Hosting using AWS S3 and Terraform

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 .|
```



The screenshot shows the Visual Studio Code interface. On the left, the Explorer pane shows a folder named 'S3STATICWEBSITE' containing a file named 'provider.tf'. The main editor area displays the contents of 'provider.tf', which is a Terraform configuration file. The file contains two main blocks: a 'terraform' block with 'required\_providers' and a 'provider' block for 'aws'.

```
1 terraform {  
2   required_providers {  
3     aws = {  
4       source = "hashicorp/aws"  
5       version = "5.88.0"  
6     }  
7   }  
8 }  
9  
10 provider "aws" {  
11   # Configuration options  
12 }
```

```
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  }
```

[terraformprojecthuerto](#)

Asia Pacific (Sydney) ap-southeast-2

[View analyzer for ap-southeast-2](#)

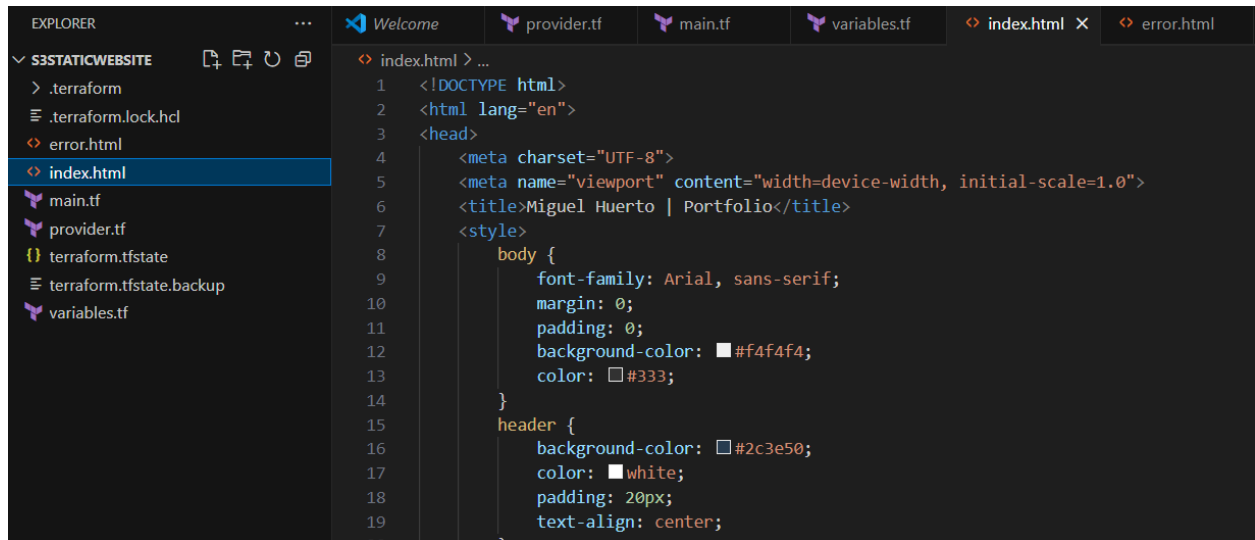
February 25, 2025, 15:40:40 (UTC+08:00)

## 2. Configure the bucket for Static Website Hosting

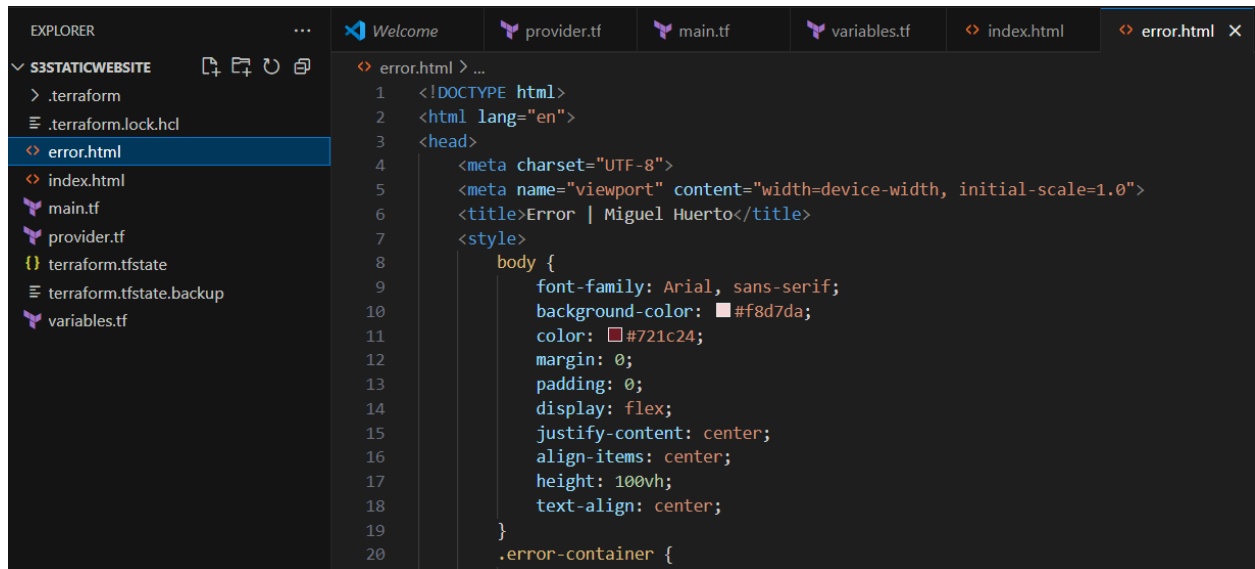
```
resource "aws_s3_bucket_ownership_controls" "example" {  
  bucket = aws_s3_bucket.mybucket.id  
  
  rule {  
    object_ownership = "BucketOwnerPreferred"  
  }  
}
```

```
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"  
}
```



```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Miguel Huerto | Portfolio</title>
7   <style>
8     body {
9       font-family: Arial, sans-serif;
10      margin: 0;
11      padding: 0;
12      background-color: #f4f4f4;
13      color: #333;
14    }
15    header {
16      background-color: #2c3e50;
17      color: white;
18      padding: 20px;
19      text-align: center;
```



```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Error | Miguel Huerto</title>
7   <style>
8     body {
9       font-family: Arial, sans-serif;
10      background-color: #f8d7da;
11      color: #721c24;
12      margin: 0;
13      padding: 0;
14      display: flex;
15      justify-content: center;
16      align-items: center;
17      height: 100vh;
18      text-align: center;
19    }
20    .error-container {
```

### 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]
}
```

[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 > ⚙

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	 <a href="#">error.html</a>	html	February 25, 2025, 16:20:46 (UTC+08:00)	1.3 KB	Standard
<input type="checkbox"/>	 <a href="#">index.html</a>	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.