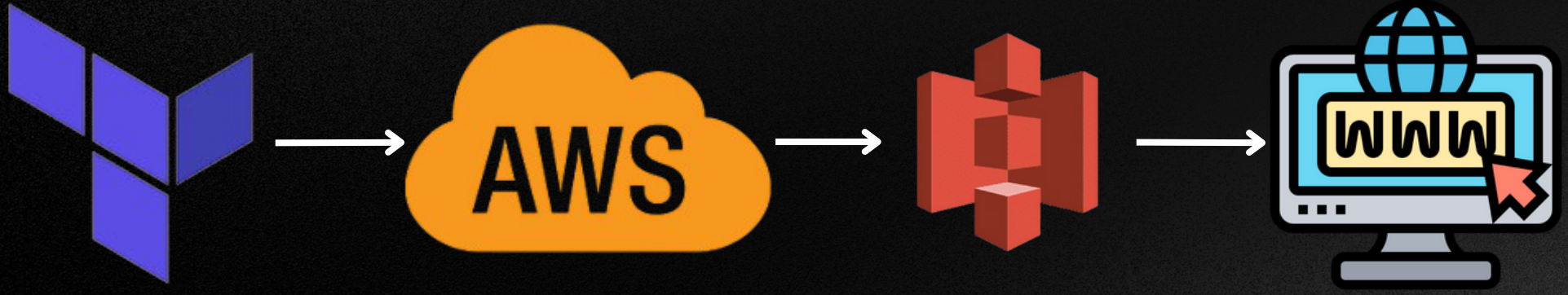
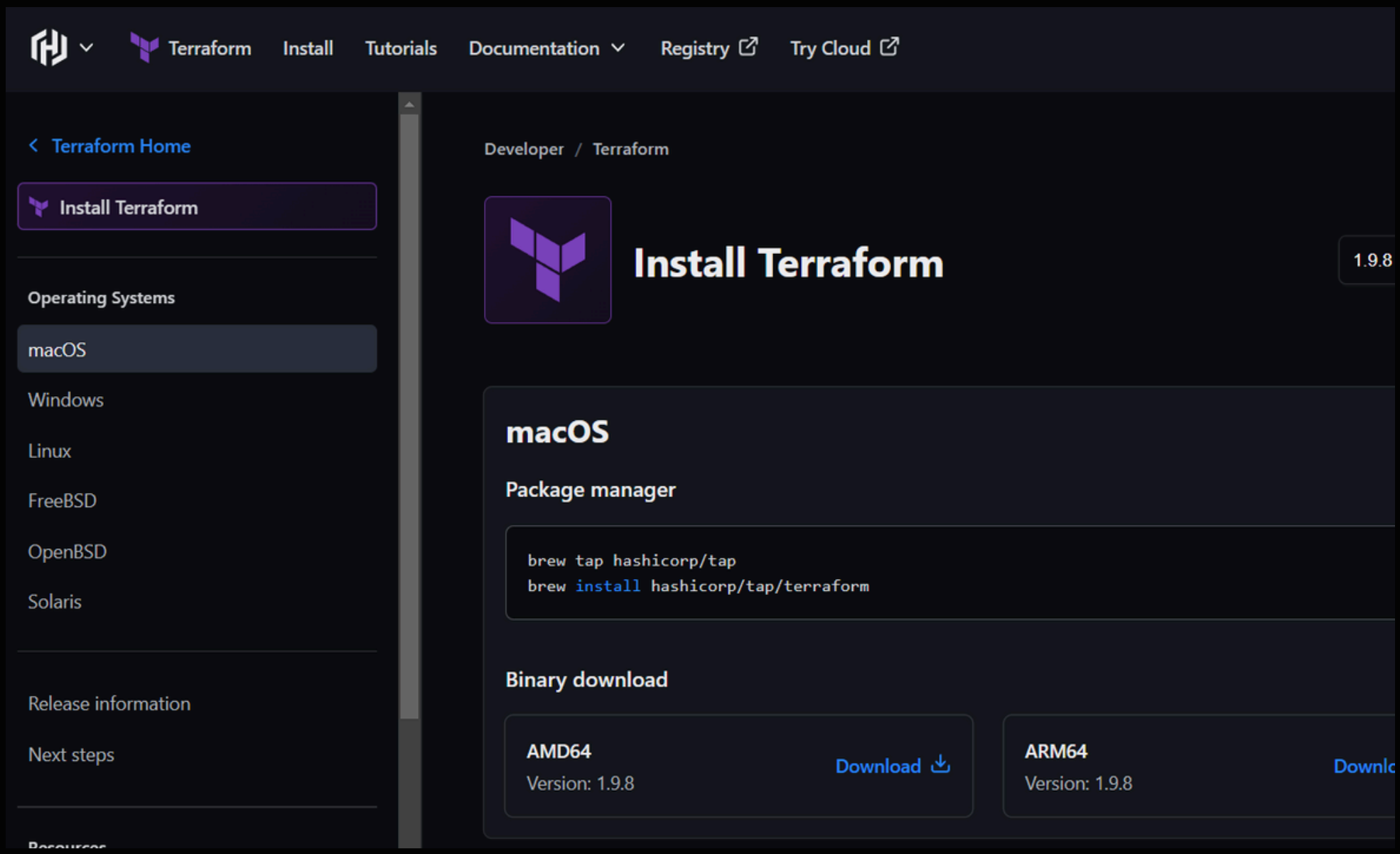
Hosting a Static Website with Terraform Automation



Overview of the Project

Hosting a static website on AWS S3 using Terraform provides an automated and efficient way to deploy and manage your web content. This project leverages Terraform, an infrastructure as code (IaC) tool, to create and configure AWS S3 buckets for hosting static website files.

Install Terraform



Create & Configure AWS Provider

provider.tf > provider "aws" > access_key

```
1   terraform {
8   }
9
10  provider "aws" {
11      region = "ap-south-1"
12      access_key = "Yourkey"
13      secret_key = "Yourkey"
14
15  }
```

Create a Variable File

provider.tf | .terraform.lock.hcl | main.tf

variable.tf > variable "bucketname"

```
1   variable "bucketname" {
2     default = "terraform-aws-staticweb-project"
3   }
```

Creation of S3 Bucket with the help of Variable

elcome | provider.tf | main.tf ×

ain.tf > resource "aws_s3_bucket" "mybucket" > buck

```
resource "aws_s3_bucket" "mybucket" {
    bucket = var.bucketname

}
```

Initialze Terraform & Validate it

```
so that Terraform can guarantee to make the same selec
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "
any changes that are required for your infrastructure.
should now work.

If you ever set or change modules or backend configurat
rerun this command to reinitialize your working directo
commands will detect it and remind you to do so if nece
```

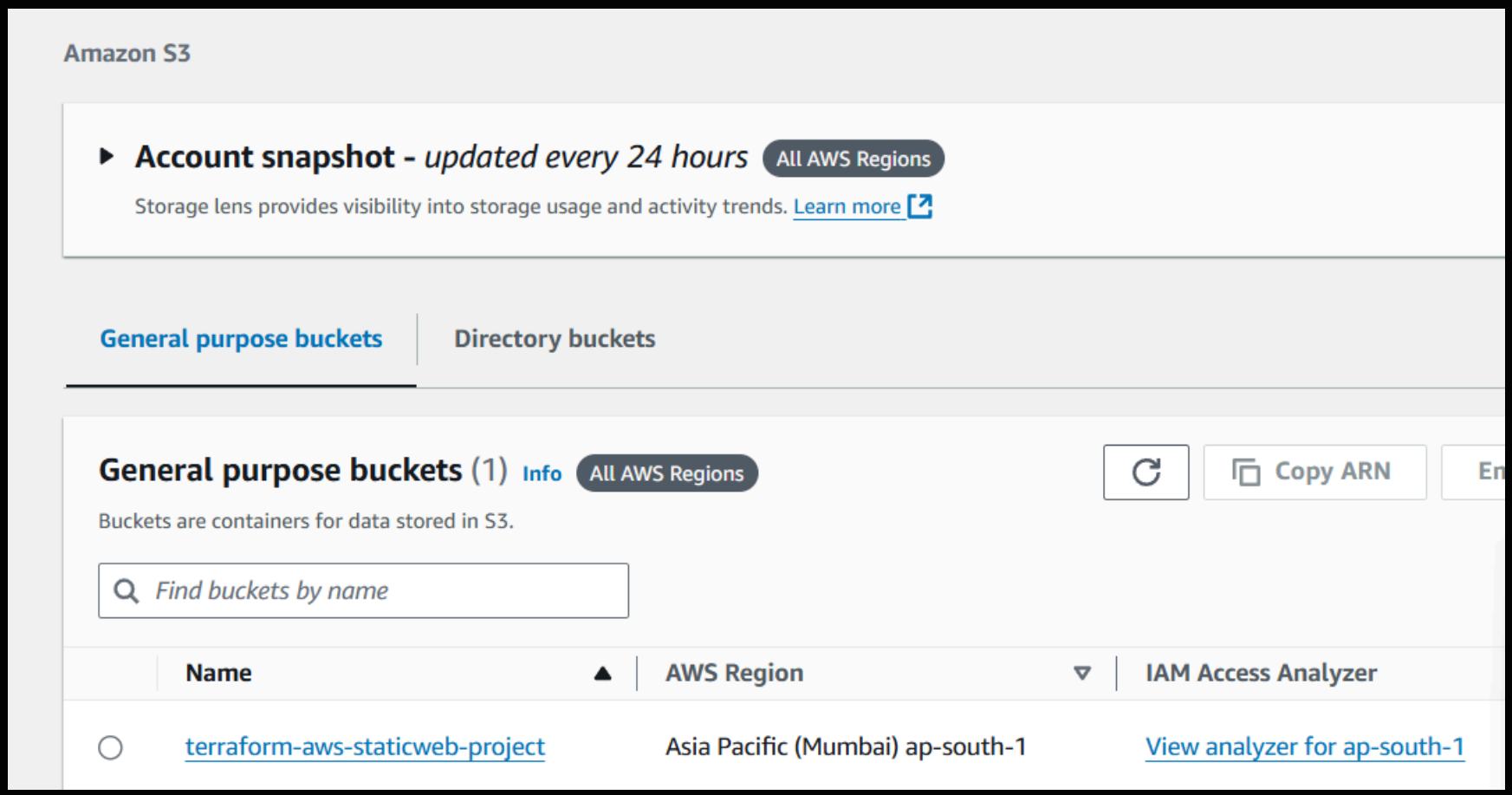
```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\kkrish\Vsc\t-final-static-web> terraform plan

Terraform used the selected providers to generate the following execut
  + create

Terraform will perform the following actions:

  # aws_s3_bucket.mybucket will be created
  + resource "aws_s3_bucket" "mybucket" {
      + acceleration_status         = (known after apply)
      + acl                         = (known after apply)
      + arn                         = (known after apply)
      + bucket                      = "terraform-aws-project"
```

Apply the Configuration

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

PS C:\Users\kkrish\Vsc\t-final-static-web> terraform apply -auto-approve

Terraform used the selected providers to generate the following execution p
  + create

Terraform will perform the following actions:

  # aws_s3_bucket.mybucket will be created
  + resource "aws_s3_bucket" "mybucket" {
      + acceleration_status         = (known after apply)
```

```
Plan: 1 to add, 0 to change, 0 to destroy.
aws_s3_bucket.mybucket: Creating...
aws_s3_bucket.mybucket: Creation complete after 1s [id=terraform-aws-staticwe

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
```

S3 Bucket


Amazon S3
Account snapshot - updated every 24 hours
All AWS Regions
Storage lens provides visibility into storage usage and activity trends. Learn more
General purpose buckets
Directory buckets
General purpose buckets (1) Info
All AWS Regions
Copy ARN
Buckets are containers for data stored in S3.
Find buckets by name
Name
AWS Region
IAM Access Analyzer
terraform-aws-staticweb-project
Asia Pacific (Mumbai) ap-south-1
View analyzer for ap-south-1

Disable ACL

provider.tf | terraform.tfstate | main.tf | variable.tf

main.tf > ...

```
resource "aws_s3_bucket" "mybucket" {
}

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"
}
```

Website Code

provider.tf | .terraform.lock.hcl | main.tf | error.html | index.html | variable.tf

index.html > html > head > style > body

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Weather App</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background-color: #f0f0f0;
            display: flex;
            flex-direction: column;
            align-items: center;
            justify-content: center;
            height: 100vh;
            margin: 0;
        }
        .weather-container {
            background: #fff;
            padding: 20px;
            border-radius: 8px;
            box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
            text-align: center;
```

Setup S3 objects of the Website code file

provider.tf | terraform.tfstate | main.tf | error.html | indexx.h

main.tf > resource "aws_s3_bucket_website_configuration" "example"

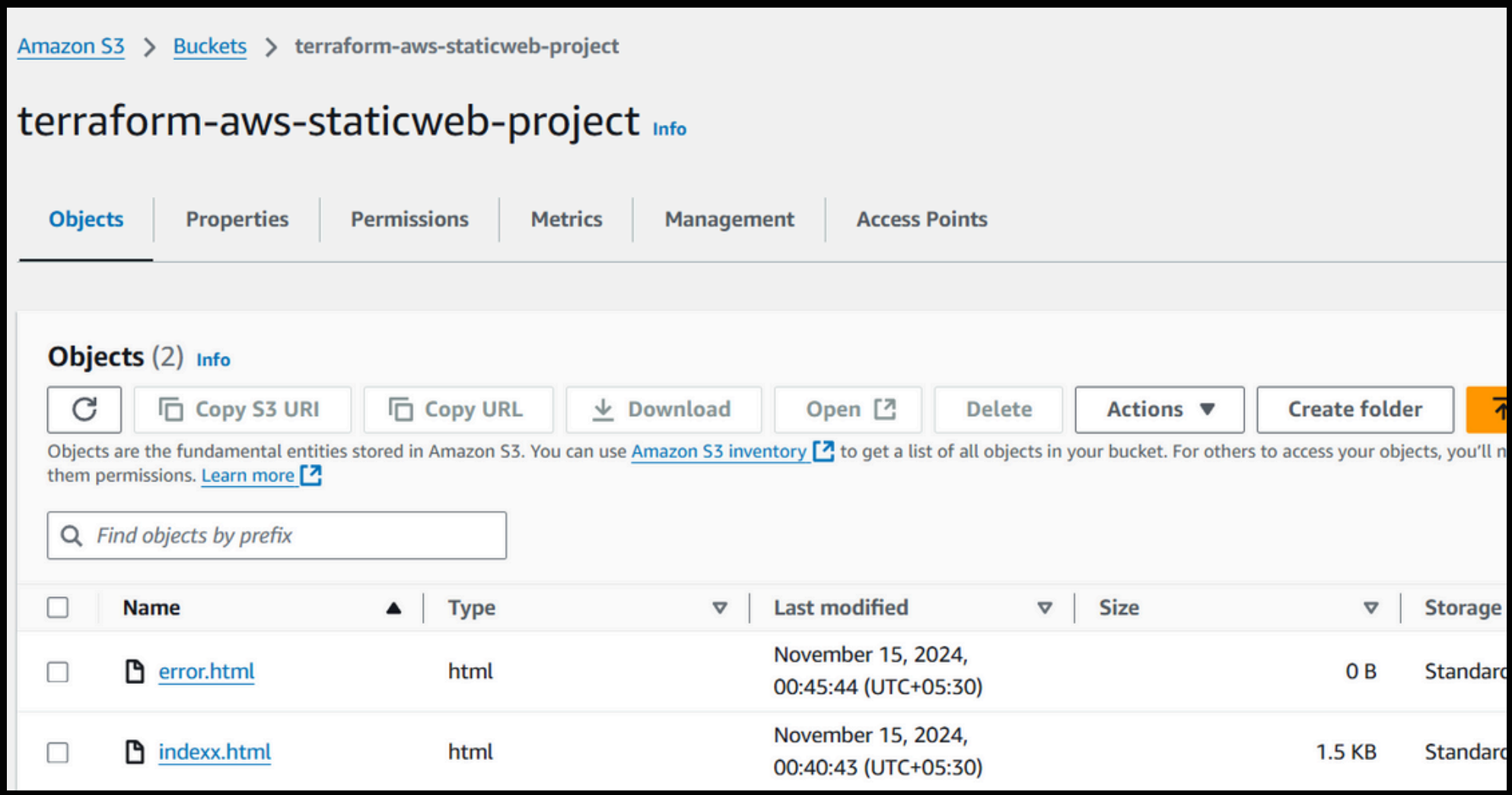
```
31
32  resource "aws_s3_object" "index" {
33    bucket = aws_s3_bucket.mybucket.id
34    key    = "indexx.html"
35    source = "indexx.html"
36    acl = "public-read"
37    content_type = "text/html"
38
39  }
40
41  resource "aws_s3_object" "error" {
42    bucket = aws_s3_bucket.mybucket.id
43    key    = "error.html"
44    source = "error.html"
45    acl = "public-read"
46    content_type = "text/html"
47
48  }
```

Upload S3 object using Terraform Apply

```
OBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

C:\Users\kkrish\Vsc\t-final-static-web> terraform apply -auto-approve
s_s3_bucket.mybucket: Refreshing state... [id=terraform-aws-staticweb-project]
s_s3_bucket_ownership_controls.example: Refreshing state... [id=terraform-aws-staticweb-pro
s_s3_bucket_public_access_block.example: Refreshing state... [id=terraform-aws-staticweb-pr
s_s3_bucket_acl.example: Refreshing state... [id=terraform-aws-staticweb-project,public-rea

rraform used the selected providers to generate the following execution plan. Resource acti
+ create
```

```
      + source                 = "indexx.html"
      + storage_class          = (known after apply)
      + tags_all               = (known after apply)
      + version_id             = (known after apply)
    }

Plan: 1 to add, 0 to change, 0 to destroy.
aws_s3_object.index: Creating...
aws_s3_object.index: Creation complete after 0s [id=indexx.html]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
```

S3 Bucket with the Website Code file



Setup Website Configuration for Web Static Hosting

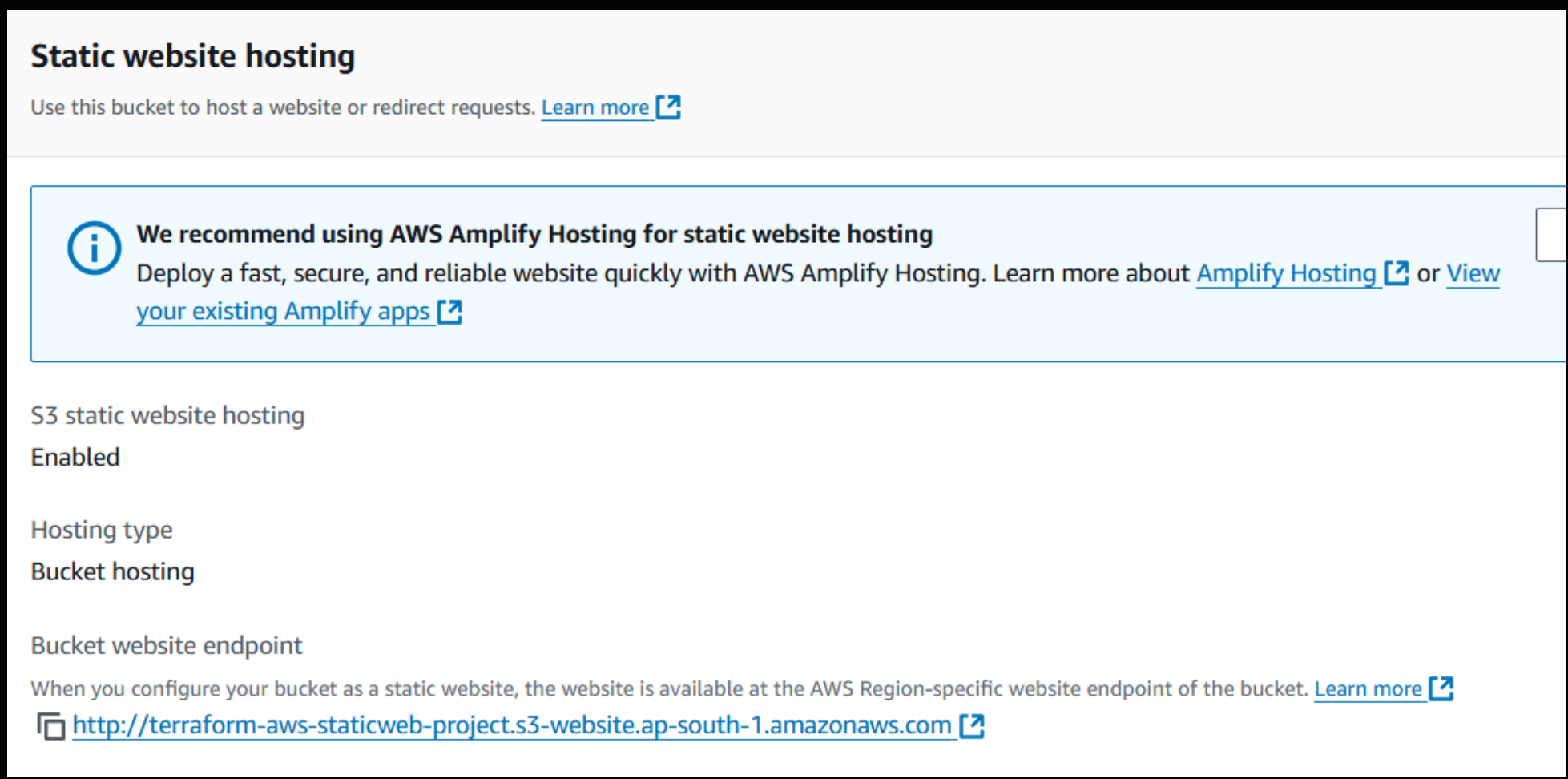
```hcl
resource "aws_s3_bucket_website_configuration" "example" {
  bucket = aws_s3_bucket.mybucket.id

  index_document {
    suffix = "index.html"
  }

  error_document {
    key = "error.html"
  }

  routing_rule {
    condition {
      key_prefix_equals = "docs/"
    }
    redirect {
      replace_key_prefix_with = "documents/"
    }
  }
}
```

Apply Website Configuration

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\kkrish\Vsc\t-final-static-web> terraform apply -auto-approve
aws_s3_bucket.mybucket: Refreshing state... [id=terraform-aws-staticweb-project]
aws_s3_bucket_public_access_block.example: Refreshing state... [id=terraform-aws-staticweb-pro
aws_s3_bucket_ownership_controls.example: Refreshing state... [id=terraform-aws-staticweb-proj
aws_s3_object.index: Refreshing state... [id=indexx.html]
aws_s3_bucket_acl.example: Refreshing state... [id=terraform-aws-staticweb-project,public-read

Terraform used the selected providers to generate the following execution plan. Resource actio
  + create

Terraform will perform the following actions:

  # aws_s3_bucket_website_configuration.example will be created
  + resource "aws_s3_bucket_website_configuration" "example" {
      + bucket           = "terraform-aws-staticweb-project"
      + id               = (known after apply)
      + routing_rules    = (known after apply)
      + website_domain   = (known after apply)
```

```
aws_s3_bucket_website_configuration.example: Creation complete aft


Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
```

Enabled Static Website hosting using Automation



Result & Outcome of the Project