* Website hosting using Terraform"

```
terraform {
required_Providers {
 aws = {
source = " hashicorp /aws"
version = " 5.54.1"
 }
 }
}

provider "aws" {
region = "~~ap-south-1~~" " ap-south-1"
 }

resource " aws_S3_bucket" " mywebapp-bkt" {
  bucket = " mywebapp-bucket"
}
```

Scanned with ACE Scanner

→ Take it from AWS provider

```
resource " aws_s3_bucket_public_access_block" "example"{

bucket = aws_s3_bucket.mywebapp-bucket.id

block_public_acls = false

block_public_policy = false

block_restrict_public_bucket = false

}
```

Take { from documen } AWS provider

```
resource "aws_s3_bucket_policy" "mywebapp" {
bucket = aws_s3_bucket.mywebapp-bucket.id

policy = jsonencode (

  {

   Version = " 2012-10-17"

   statement = [

    {

    sid      = " publicReadGetobject",

    effect = " Allow",

    Principal = " * ",

    Action = " s3: Getobject",

    Resource = " arn: aws: s3 : : : ${aws_s3_bkt. mywebapp-bkt
                                                .id}/*"

    }

  ]

}
```

Scanned with ACE Scanner

```
resource "aws_s3_object" "index.html"{
bucket = aws_s3_bucket.mywebapp_bucket.bucket
source = "./index.html"
key = "index.html"
content_type = "text/html"

}

resource "aws_s3_object" "styles_css" {
bucket = aws_s3_bucket.mywebapp_bucket.bucket
source = "./styles.css"
key = "styles.css"
content_type = "styles.css"

}

output "name" {
value = aws_s3_bucket_website_configuration.mywebapp_
                                    website_endpoint

}
```

Scanned with ACE Scanner

```
terraform {
  required_providers {
    aws = {
      source  = "hashicorp/aws"
      version = "5.54.1"
    }
    random = {
      source  = "hashicorp/random"
      version = "3.6.2"
    }
  }
}

provider "aws" {
  region = "eu-north-1"
}
```

Scanned with ACE Scanner

```
provider "aws" {
  region = "eu-north-1"
}

resource "random_id" "rand_id" {…
}

resource "aws_s3_bucket" "demo-bucket" {
  bucket = "demo-bucket-${random_id.rand_id.hex}"
}

resource "aws_s3_object" "bucket-data" {
  bucket = aws_s3_bucket.demo-bucket.bucket
  source = "./myfile.txt"
  key    = "mydata.txt"
}
```

Scanned with ACE Scanner

```
resource "aws_s3_object" "index_html" {
  bucket       = aws_s3_bucket.mywebapp-bucket.bucket
  source       = "./index.html"
  key          = "index.html"
  content_type = "text/html"
}

resource "aws_s3_object" "styles_css" {
  bucket       = aws_s3_bucket.mywebapp-bucket.bucket
  source       = "./styles.css"
  key          = "styles.css"
  content_type = "text/css"
}

output "name" {
```

Scanned with ACE Scanner

```
resource "aws_s3_bucket_public_access_block" "example" {
  bucket = aws_s3_bucket.mywebapp-bucket.id

  block_public_acls       = false
  block_public_policy     = false
  ignore_public_acls      = false
  restrict_public_buckets = false
}


resource "aws_s3_object" "index_html" {
  bucket = aws_s3_bucket.mywebapp-bucket.bucket
  source = "./index.html"
  key    = "index.html"
}
```

terraform - proj-st

Scanned with ACE Scanner

```
bucket = aws_s3_bucket.mywebapp-bucket.id
policy = jsonencode(
  {
    Version = "2012-10-17",
    Statement = [
      {
        Sid       = "PublicReadGetObject",
        Effect    = "Allow",
        Principal = "*",
        Action    = "s3:GetObject",
        Resource  = "arn:aws:s3:::${aws_s3_bucket.mywebapp-bucket.id}/*"
      }
    ]
  }
)
```

Scanned with ACE Scanner

```
resource "aws_s3_object" "index_html" {
  bucket = aws_s3_bucket.mywebapp-bucket.bucket
  source = "./index.html"
  key    = "index.html"
}

resource "aws_s3_object" "styles_css" {
  bucket = aws_s3_bucket.mywebapp-bucket.bucket
  source = "./styles.css"
  key    = "styles.css"
}

output "name" {
  value = aws_s3_bucket_website_configuration.mywebapp.website_endpoint
}
```

Scanned with ACE Scanner