

### **SCENARIO:-**

Users within a company need a place to store their documents. Each user must have his/her own location for placing the set of documents and should not be able to view another person's documents. Also, users should be able to retrieve their documents easily. Implement the AWS Service and Resources to achieve this objective.

Configure and set up S3 Bucket policy user wise and role wise for this organization.

### **Lab Overview:-**

This lab focuses on Amazon S3 or Amazon Simple Storage Service offered by Amazon Web Services that provides object storage through a web service interface. Amazon S3 uses the same scalable storage infrastructure that Amazon.com uses to run its global e-commerce network. You are required to accomplish the following task based on the scenarios as shown in figure:

Refer to the lab scenario attached herewith. Tasks to be completed:

- Create the S3 bucket.
- Give the public access to S3 buckets.
- Host static HTML site from S3 buckets.
- Life-Cycle Policy in S3 Bucket.

**AIM:-** Develop a Terraform script to host static HTML websites from s3 buckets.

### **Steps:-**

#### **1. Configuring terraform script.**

```
- CODE:-
terraform {
  required_providers {
    aws = {
      source  = "hashicorp/aws"
      version = "~> 3.0"
    }
  }
  required_version = ">= 0.13.5"
}

provider "aws" {
  region      = "us-east-1"
  access_key  = "YOUR_ACCESS_KEY"
  secret_key  = "YOUR_SECRET_KEY"
}

variable "bucket_name" {
  default = "e4-demo-bucket"
}
```

```
resource "aws_s3_bucket" "buck" {
  bucket = var.bucket_name
  acl    = "public-read"

  policy = <<EOF
{
  "Id": "bucket_policy_site",
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "bucket_policy_site_main",
      "Action": [
        "s3:PutObject",
        "s3:PutObjectAcl"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::${var.bucket_name}/*",
      "Principal": "*"
    }
  ]
}
EOF

website {
  index_document = "index.html"
  error_document = "error.html"
}

resource "aws_s3_bucket_object" "index_ob" {
  bucket      = aws_s3_bucket.buck.bucket
  key         = "index.html"
  source      = "index.html"
  acl         = "public-read"
  content_type = "text/html"
}

resource "aws_s3_bucket_object" "error_ob" {
  bucket      = aws_s3_bucket.buck.bucket
  key         = "error.html"
  source      = "error.html"
  acl         = "public-read"
  content_type = "text/html"
}
```

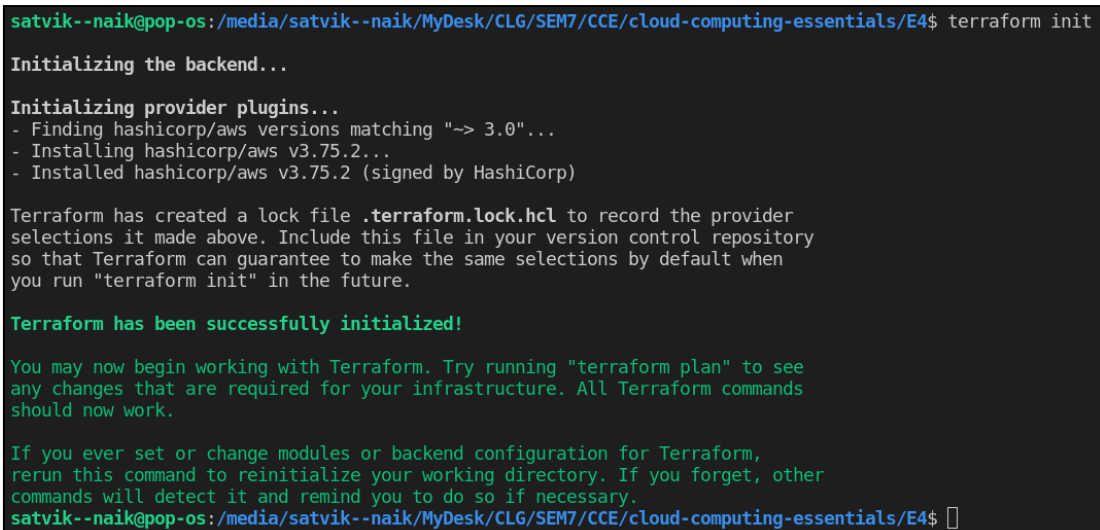
```
resource "aws_s3_bucket_lifecycle_configuration" "buck-lifecycle-config" {
  bucket = aws_s3_bucket.buck.id
  rule {
    id = "rule-1"
    expiration {
      days = 1
    }
    status = "Enabled"
  }
}

output "website_domain" {
  value = aws_s3_bucket.buck.website_domain
}

output "website_endpoint" {
  value = aws_s3_bucket.buck.website_endpoint
}
```

## 2. Launch terraform script.

- terraform init



```
satvik--naik@pop-os: /media/satvik--naik/MyDesk/CLG/SEM7/CCE/cloud-computing-essentials/E4$ terraform init

Initializing the backend...

Initializing provider plugins...
- Finding hashicorp/aws versions matching ">= 3.0"...
- Installing hashicorp/aws v3.75.2...
- Installed hashicorp/aws v3.75.2 (signed by HashiCorp)

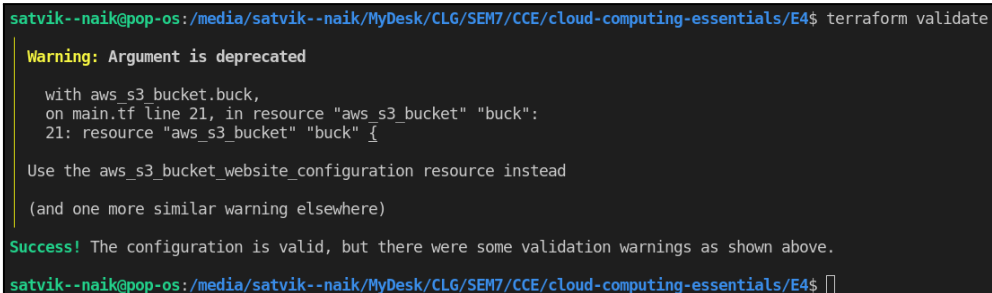
Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
satvik--naik@pop-os: /media/satvik--naik/MyDesk/CLG/SEM7/CCE/cloud-computing-essentials/E4$
```

- terraform validate



```
satvik--naik@pop-os: /media/satvik--naik/MyDesk/CLG/SEM7/CCE/cloud-computing-essentials/E4$ terraform validate

Warning: Argument is deprecated

  with aws_s3_bucket.buck,
  on main.tf line 21, in resource "aws_s3_bucket" "buck":
  21: resource "aws_s3_bucket" "buck" {

Use the aws_s3_bucket_website_configuration resource instead

(and one more similar warning elsewhere)

Success! The configuration is valid, but there were some validation warnings as shown above.
satvik--naik@pop-os: /media/satvik--naik/MyDesk/CLG/SEM7/CCE/cloud-computing-essentials/E4$
```

- terraform plan

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

Plan: 4 to add, 0 to change, 0 to destroy.

Changes to Outputs:
+ website_domain   = (known after apply)
+ website_endpoint = (known after apply)

Warning: Argument is deprecated

  with aws_s3_bucket.buck,
  on main.tf line 21, in resource "aws_s3_bucket" "buck":
  21: resource "aws_s3_bucket" "buck" {

Use the aws_s3_bucket_website_configuration resource instead

(and 3 more similar warnings elsewhere)

Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if you run
"terraform apply" now.
satvik--naik@pop-os: /media/satvik--naik/MyDesk/CLG/SEM7/CCE/cloud-computing-essentials/E4$
```

- terraform apply -auto-approve

```
aws_s3_bucket.buck: Creation complete after 16s [id=e4-demo-bucket]
aws_s3_bucket_lifecycle_configuration.buck-lifecycle-config: Creating...
aws_s3_bucket_object.index_ob: Creating...
aws_s3_bucket_object.error_ob: Creating...
aws_s3_bucket_object.error_ob: Creation complete after 3s [id=error.html]
aws_s3_bucket_object.index_ob: Creation complete after 3s [id=index.html]
aws_s3_bucket_lifecycle_configuration.buck-lifecycle-config: Still creating... [10s elapsed]
aws_s3_bucket_lifecycle_configuration.buck-lifecycle-config: Still creating... [20s elapsed]
aws_s3_bucket_lifecycle_configuration.buck-lifecycle-config: Still creating... [30s elapsed]
aws_s3_bucket_lifecycle_configuration.buck-lifecycle-config: Creation complete after 36s [id=e4-demo-bucket]

Warning: Argument is deprecated

  with aws_s3_bucket.buck,
  on main.tf line 21, in resource "aws_s3_bucket" "buck":
  21: resource "aws_s3_bucket" "buck" {

Use the aws_s3_bucket_website_configuration resource instead

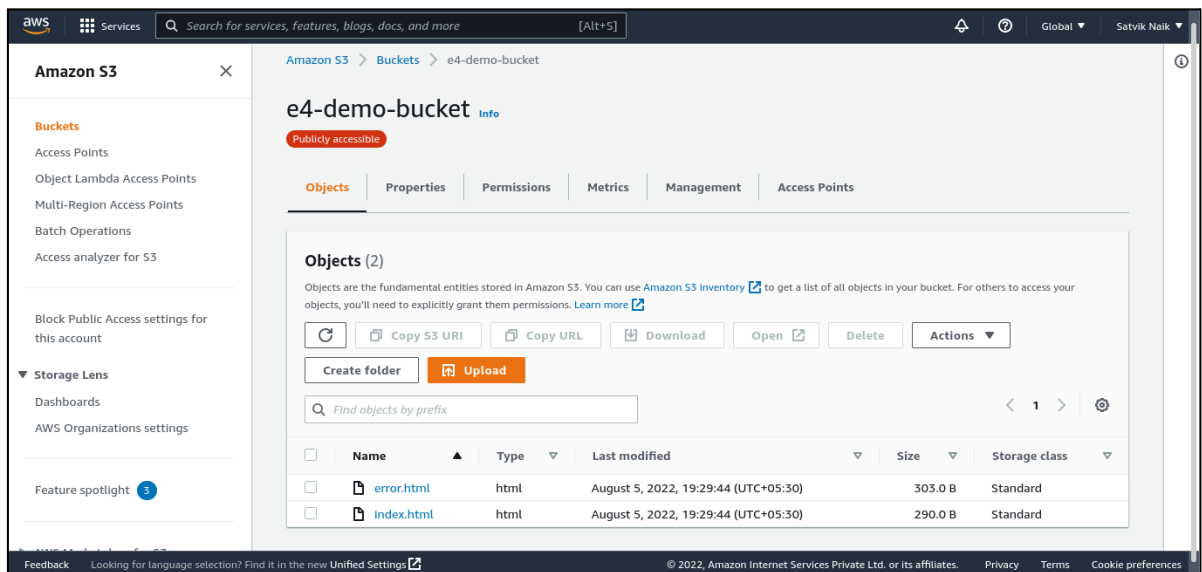
(and 5 more similar warnings elsewhere)

Apply complete! Resources: 4 added, 0 changed, 0 destroyed.

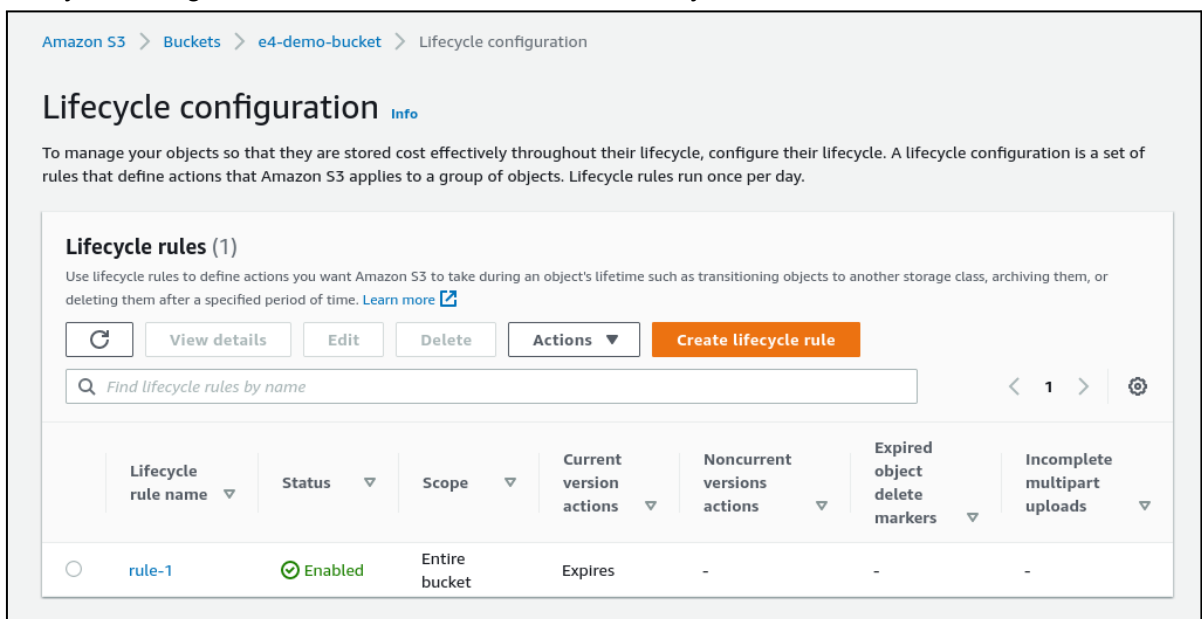
Outputs:
website_domain = "s3-website-us-east-1.amazonaws.com"
website_endpoint = "e4-demo-bucket.s3-website-us-east-1.amazonaws.com"
satvik--naik@pop-os: /media/satvik--naik/MyDesk/CLG/SEM7/CCE/cloud-computing-essentials/E4$
```

### 3. Verifying resource created on AWS Cloud.

- S3 bucket created successfully.



- Lifecycle configuration created & enabled successfully.



- Access the hosted website.

