

**Mohit Shadija**

**D15B**

**54**

## **Adv Devops Assignment 2**

### **Code:**

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

# S3 Bucket
resource "aws_s3_bucket" "s3mayur" {
  bucket = "my-terraform-s3-bucket"
  acl    = "private"

  versioning {
    enabled = true
  }
}

# SQS Queue
resource "aws_sqs_queue" "sqsmayur" {
  name = "my-terraform-sqs-queue"
}

# Lambda Function
resource "aws_lambda_function" "lambda_mayur" {
  function_name = "s3-to-sqs-lambda"
  role          = aws_iam_role.lambda_exec.arn
  handler       = "index.handler"
  runtime       = "nodejs14.x"
  timeout       = 10

  filename = "lambda.zip" # Path to the Lambda zip file

  environment {
    variables = {
      QUEUE_URL = aws_sqs_queue.sqsmayur.id
    }
  }
}

# IAM Role for Lambda execution
resource "aws_iam_role" "lambda_exec" {
```

```

name = "lambda_exec_role"

assume_role_policy = jsonencode({
  Version = "2012-10-17",
  Statement = [{
    Action   = "sts:AssumeRole",
    Effect    = "Allow",
    Principal = {
      Service = "lambda.amazonaws.com"
    }
  }]
})
}

# IAM Role Policy for Lambda (grant permissions to interact with S3 and SQS)
resource "aws_iam_role_policy" "lambda_exec_policy" {
  role = aws_iam_role.lambda_exec.id

  policy = jsonencode({
    Version = "2012-10-17",
    Statement = [
      {
        Action = [
          "sqs:SendMessage"
        ],
        Effect  = "Allow",
        Resource = aws_sqs_queue.sqsmayur.arn
      },
      {
        Action = [
          "s3:GetObject"
        ],
        Effect  = "Allow",
        Resource = "${aws_s3_bucket.s3mayur.arn}/*"
      }
    ]
  })
}

# S3 Bucket Notification to trigger Lambda on object creation
resource "aws_s3_bucket_notification" "s3_notification" {
  bucket = aws_s3_bucket.s3mayur.id

  lambda_function {

```

```

    lambda_function_arn = aws_lambda_function.lambda_mayur.arn
    events                = ["s3:ObjectCreated:*"]
  }
}

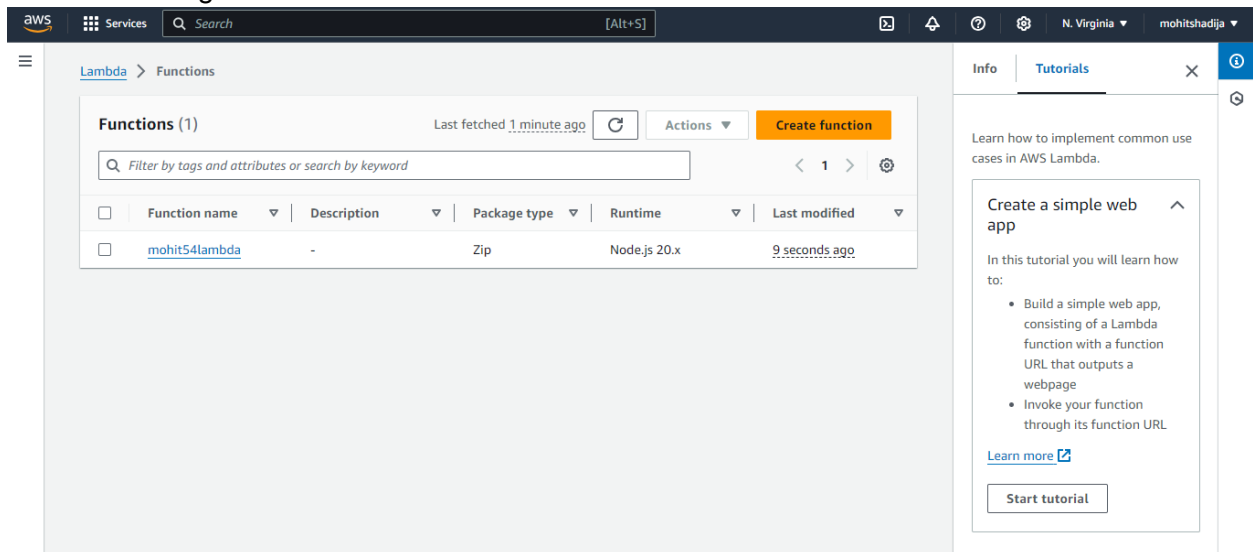
# Lambda Permission for S3 to invoke the Lambda function
resource "aws_lambda_permission" "allow_s3" {
  statement_id = "AllowS3InvokeLambda"
  action       = "lambda:InvokeFunction"
  function_name = aws_lambda_function.lambda_mayur.function_name
  principal     = "s3.amazonaws.com"

  source_arn = aws_s3_bucket.s3mayur.arn
}

```

## Implementation:

### 1. Creating Lambda Function



The screenshot shows the AWS Lambda console interface. The top navigation bar includes the AWS logo, 'Services' menu, a search bar, and user information for 'N. Virginia' and 'mohitshadija'. The main content area is titled 'Lambda > Functions'. It displays a table with one function: 'mohit54lambda'. The table columns are 'Function name', 'Description', 'Package type', 'Runtime', and 'Last modified'. The function 'mohit54lambda' has a description of '-', package type of 'Zip', runtime of 'Node.js 20.x', and was last modified '9 seconds ago'. Above the table, there is a 'Functions (1)' header, a 'Last fetched 1 minute ago' timestamp, a refresh button, an 'Actions' dropdown, and a 'Create function' button. A search bar is also present with the placeholder text 'Filter by tags and attributes or search by keyword'. On the right side, there is a 'Tutorials' sidebar with a 'Create a simple web app' tutorial card. The card includes a brief description, a list of steps, a 'Learn more' link, and a 'Start tutorial' button.

Function name	Description	Package type	Runtime	Last modified
mohit54lambda	-	Zip	Node.js 20.x	9 seconds ago

### 2. Creating Sqs Queue

Services Search [Alt+S] N. Virginia mohitshadija

Amazon SQS > Queues

Queues (1) Edit Delete Send and receive messages Actions Create queue

Search queues by prefix < 1 >

Name	Type	Created	Messages available	Messages in flight	Encryption	Content-based
Mohit54	Standard	2024-10-17T22:50+05:30	0	0	Amazon SQS key (SSE-SQS)	-

### 3. Creating S3 Bucket

Services Search N. Virginia mohitshadija

Amazon S3 > Buckets > mohit54s3

mohit54s3 Info

Objects Properties Permissions Metrics Management Access Points

Objects (0) Info

Copy S3 URI Copy URL Download Open Delete

Actions Create folder Upload

Performing Terraform commands

## 1. Terraform init

```
PS C:\Users\sujal\Documents\terraform-aws-s3-sqs-lambda> terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.72.1...
- Installed hashicorp/aws v5.72.1 (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.
```

## 2. Terraform plan

```
PS C:\Users\sujal\Documents\terraform-aws-s3-sqs-lambda> terraform plan
```

```
Warning: Argument is deprecated

with aws_s3_bucket.s3sujal,
on main.tf line 6, in resource "aws_s3_bucket" "s3sujal":
6: resource "aws_s3_bucket" "s3sujal" {

Use the aws_s3_bucket_versioning resource instead

(and one more similar warning elsewhere)
```

## 3. Terraform apply

```
PS C:\Users\sujal\Documents\terraform-aws-s3-sqs-lambda> terraform apply
```

**Warning:** Argument is deprecated

```
with aws_s3_bucket.s3sujal,  
on main.tf line 6, in resource "aws_s3_bucket" "s3sujal":  
6: resource "aws_s3_bucket" "s3sujal" {
```

Use the `aws_s3_bucket_versioning` resource instead

(and one more similar warning elsewhere)

#### 4. Terraform destroy

```
PS C:\Users\sujal\Documents\terraform-aws-s3-sqs-lambda> terraform destroy
```

**Warning:** Argument is deprecated

```
with aws_s3_bucket.s3sujal,  
on main.tf line 6, in resource "aws_s3_bucket" "s3sujal":  
6: resource "aws_s3_bucket" "s3sujal" {
```

Use the `aws_s3_bucket_versioning` resource instead

(and one more similar warning elsewhere)

Folder structure of main.tf file

EXPLORER

✓ TERRAFORM-AWS-S3-SQS-LAMBDA

> .terraform

≡ .terraform.lock.hcl

≡ .terraform.tfstate.lock.info

main.tf

≡ terraform.tfstate

### Conclusion:

In this experiment, we successfully deployed an AWS infrastructure using Terraform, integrating essential services such as Amazon S3, SQS, and Lambda. By leveraging Terraform's infrastructure as code capabilities, we were able to automate the provisioning and configuration of cloud resources, ensuring consistency and reproducibility in our deployments.