

AWS Lambda Function

The screenshot shows the AWS Lambda console for a function named 'myLambdaFunction'. The interface includes a top navigation bar with the AWS logo, a search bar, and user information. The main content area is titled 'myLambdaFunction' and features a 'Function overview' section with tabs for 'Diagram' and 'Template'. The 'Diagram' tab shows a visual representation of the function's architecture, including a 'myLambdaFunction' box connected to an 'API Gateway' box. A '+ Add trigger' button is visible below the API Gateway. To the right of the diagram, there is a 'Description' section with fields for 'Last modified' (14 minutes ago), 'Function ARN' (arn:aws:lambda:us-east-2:842676015714:function:myLambdaFunction), and 'Function URL' (https://s4x5q5h4neyvyiuarmaa5qju0doubu.lambda-url-us-east-2.on.aws/). Below the diagram, there are tabs for 'Code', 'Test', 'Monitor', 'Configuration', 'Aliases', and 'Versions'. The 'Test' tab is currently selected, showing a 'Test event' section with a 'Test' button and a 'CloudWatch Logs Live Tail' button.

DynamoDB table with data uploaded by Lambda

The screenshot shows the AWS DynamoDB console for a table named 'MIUWorkshop3Table'. The interface includes a top navigation bar with the AWS logo, a search bar, and user information. The main content area is titled 'MIUWorkshop3Table' and features a 'Scan or query items' section with buttons for 'Scan' and 'Query'. Below this, there are dropdown menus for 'Select a table or index' (Table - MIUWorkshop3Table) and 'Select attribute projection' (All attributes). A 'Run' button is visible. Below the 'Scan or query items' section, there is a green status bar indicating 'Completed · Items returned: 1 · Items scanned: 1 · Efficiency: 100% · RCUs consumed: 0.5'. Below this, there is a section titled 'Table: MIUWorkshop3Table - Items returned (1)' with a 'Create item' button. The table shows one item with columns for 'Email (String)', 'Datetime', and 'URL'. The data row shows 'gauravneupane.tech...', '2025-04-1...', and 'https://my-image-bucket-miu-workshop3.s3.amazonaws.com/...'.

SNS topic for getting message

The screenshot shows the Amazon SNS console interface. On the left is a navigation menu with 'Amazon SNS', 'Topics', 'Subscriptions', and 'Mobile' sections. The main content area displays the details for the topic 'my-notification-topic'. A blue banner at the top indicates a new feature: 'Amazon SNS now supports High Throughput FIFO topics. Learn more'. Below the banner, the topic name is 'my-notification-topic' with buttons for 'Edit', 'Delete', and 'Publish message'. The 'Details' section shows the Name, Display name, ARN, Topic owner, and Type. Below this is a tabbed interface with 'Subscriptions' selected, showing a table of subscriptions. The table has columns for ID, Endpoint, Status, and Protocol. One subscription is listed with ID '45a58d8e-64bc-4b95-b73e-607a...', Endpoint 'gauravneupane.tech@gmail.com', Status 'Confirmed', and Protocol 'EMAIL'.

Amazon SNS

Dashboard
Topics
Subscriptions

▼ **Mobile**
Push notifications
Text messaging (SMS)

my-notification-topic [Edit] [Delete] [Publish message]

Details

Name my-notification-topic	Display name -
ARN arn:aws:sns:us-east-2:842676015714:my-notification-topic	Topic owner 842676015714
Type Standard	

< **Subscriptions** | Access policy | Data protection policy | Delivery policy (HTTP/S) | Delivery status logging | Encryption | Tag >

Subscriptions (1) [Edit] [Delete] [Request confirmation] [Confirm subscription] [Create subscription]

Search

ID	Endpoint	Status	Protocol
45a58d8e-64bc-4b95-b73e-607a...	gauravneupane.tech@gmail.com	Confirmed	EMAIL

Got sns email by sending message through lambda

The screenshot shows a Gmail inbox. The left sidebar has 'Compose', 'Inbox' (2,485), 'Starred', 'Snoozed', 'Sent', 'More', and 'Labels' (jobs_listings 2,325, Jobs_notifications). The main email is from 'AWS Notifications' to 'me', received at 12:07 AM. The subject is 'AWS Notification Message'. The body text says 'Data uploaded successfully' and includes an unsubscribe link: 'https://sns.us-east-2.amazonaws.com/unsubscribe.htm?SubscriptionArn=arn:aws:sns:us-east-2:842676015714:my-notification-topic:45a58d8e-64bc-4b95-b73e-607a99deb181&Endpoint=gauravneupane.tech@gmail.com'. It also mentions a support link: 'https://aws.amazon.com/support'.

Gmail

Compose

Inbox 2,485

Starred
Snoozed
Sent
More

Labels +

jobs_listings 2,325
Jobs_notifications

AWS Notification Message [Inbox x]

AWS Notifications to me 12:07 AM (0 minutes ago)

Data uploaded successfully

--

If you wish to stop receiving notifications from this topic, please click or visit the link below to unsubscribe:
<https://sns.us-east-2.amazonaws.com/unsubscribe.htm?SubscriptionArn=arn:aws:sns:us-east-2:842676015714:my-notification-topic:45a58d8e-64bc-4b95-b73e-607a99deb181&Endpoint=gauravneupane.tech@gmail.com>

Please do not reply directly to this email. If you have any questions or comments regarding this email, please contact us at <https://aws.amazon.com/support>

Cloud front for frontend URL

The screenshot shows the AWS CloudFront console for distribution EHOKLLUWKGPCPK. The 'General' tab is selected, displaying details, settings, and deployment options.

Details

- Distribution domain name: d33fp0hgkwl3x5.cloudfront.net
- ARN: arn:aws:cloudfront:842676015714:distribution/EHOKLLUWKGPCPK
- Last modified: April 17, 2025 at 3:54:23 AM UTC

Settings

- Description: -
- Alternate domain names: -
- Standard logging: Off
- Cookie logging: Off
- Default root object: index.html
- Price class: Use all edge locations (best performance)
- Supported HTTP versions: HTTP/2, HTTP/1.1, HTTP/1.0

Continuous deployment [info](#)

[Create staging distribution](#)

S3 for React build

The screenshot shows the AWS S3 console for the bucket 'miu-workshop-3-frontend-react-bck'. The 'Objects' tab is selected, displaying a list of objects.

Objects (8)

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

Name	Type	Last modified	Size	Storage class
asset-manifest.json	json	April 17, 2025, 23:49:48 (UTC-05:00)	517.0 B	Standard
favicon.ico	ico	April 17, 2025, 23:49:49 (UTC-05:00)	3.8 KB	Standard
index.html	html	April 17, 2025, 23:49:49 (UTC-05:00)	644.0 B	Standard
logo192.png	png	April 17, 2025, 23:49:49 (UTC-05:00)	5.2 KB	Standard
logo512.png	png	April 17, 2025, 23:49:49 (UTC-05:00)	9.4 KB	Standard
manifest.json	json	April 17, 2025, 23:49:49 (UTC-05:00)	492.0 B	Standard
robots.txt	txt	April 17, 2025, 23:49:49 (UTC-05:00)	67.0 B	Standard
static/	Folder	-	-	-

S3 for uploading images

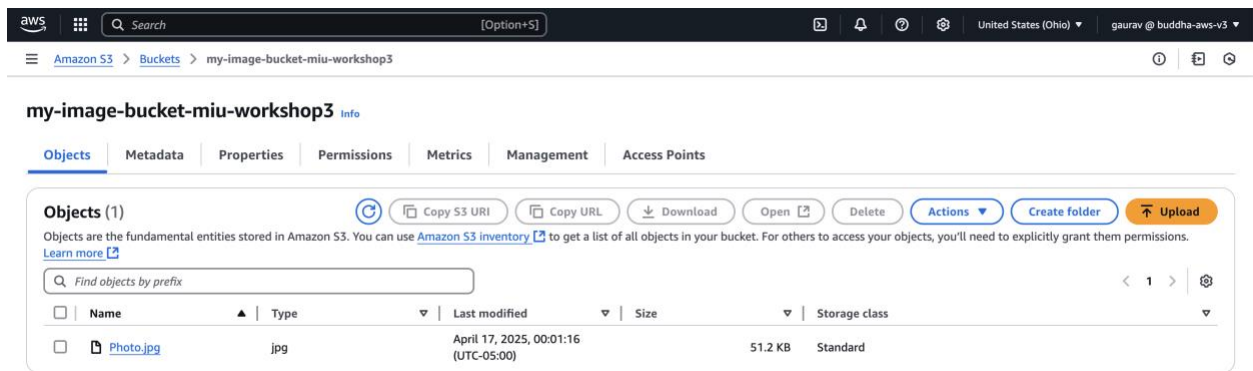
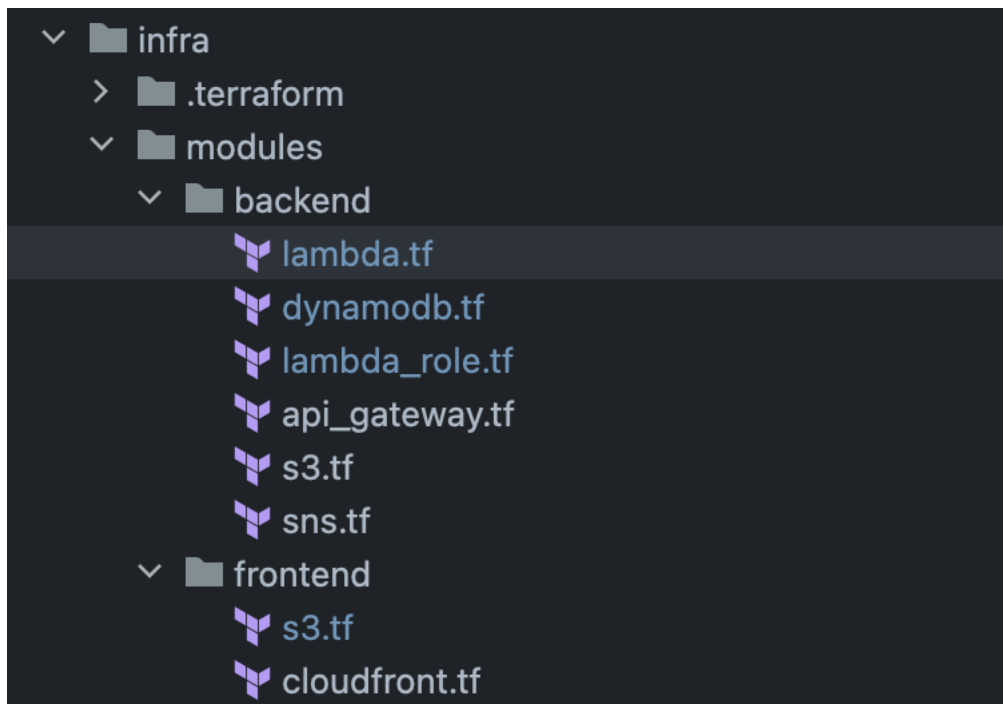


Image uploaded successfully.



Terraform files hosting backend and frontend using IAC.



Using Terraform to manage the cloud resources

```
terraform {  
  required_providers {  
    aws = {  
      source  = "hashicorp/aws"  
      version = "~> 5.0"  
    }  
  }  
}  
  
# Configure the AWS Provider  
provider "aws" {  
  region = "us-east-2"  
}  
  
module "backend"{  
  source = "./modules/backend"  
}  
  
module "frontend"{  
  source = "./modules/frontend"  
}
```

CORS for S3

```
resource "aws_s3_bucket_cors_configuration" "miu_bucket_cors" {
  bucket = aws_s3_bucket.image_s3.id

  cors_rule {
    allowed_headers = ["*"]
    allowed_methods = ["PUT", "POST"]
    allowed_origins = ["*"]
    expose_headers  = ["ETag"]
    max_age_seconds = 3000
  }
}
```

CORS for AWS gateway

```
resource "aws_apigatewayv2_api" "lambda_api" {
  name          = "lambda-http-api"
  protocol_type = "HTTP"

  cors_configuration {
    allow_headers = ["Content-Type"]
    allow_methods = ["POST"]
    allow_origins = ["*"]
    expose_headers = []
    max_age       = 3600
  }
}
```