

AWS Lambda

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 divided into sections: 'Function overview' with tabs for 'Diagram' and 'Template', a 'Diagram' showing the function connected to an 'API Gateway', and a 'Description' section with details like 'Last modified', 'Function ARN', and 'Function URL'. There are buttons for 'Throttle', 'Copy ARN', 'Actions', 'Export to Infrastructure Composer', and 'Download'. Below the overview, there are tabs for 'Code', 'Test', 'Monitor', 'Configuration', 'Aliases', and 'Versions'. The 'Test' tab is active, showing a 'Test event' section with a 'Test' 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 divided into sections: a left sidebar with navigation links, a 'Tables (1)' section showing the table, and a 'Scan or query items' section. The 'Scan or query items' section has buttons for 'Scan' and 'Query', a 'Run' button, and a 'Reset' button. Below this, there is a 'Completed' status bar showing 'Items returned: 1', 'Items scanned: 1', 'Efficiency: 100%', and 'RCUs consumed: 0.5'. At the bottom, there is a table titled 'Table: MIUWorkshop3Table - Items returned (1)' with columns for 'Email (String)', 'Datetime', and 'URL'. The table contains one row of data.

Email (String)	Datetime	URL
gauravneupane.tech...	2025-04-1...	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 a topic named '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 'my-notification-topic' is shown with 'Edit', 'Delete', and 'Publish message' buttons. The 'Details' section contains a table with the following information:

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

Below the details, there are tabs for 'Subscriptions', 'Access policy', 'Data protection policy', 'Delivery policy (HTTP/S)', 'Delivery status logging', 'Encryption', and 'Tag'. The 'Subscriptions' tab is active, showing a table with one subscription:

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 with an email from 'AWS Notifications'. The email subject is 'AWS Notification Message'. The sender is 'AWS Notifications' and it was received '12:07 AM (0 minutes ago)'. The email content states '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 includes a support link: <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 the following details:

- Details:**
 - Distribution domain name: d33fp0hqkwl3x5.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`
- 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 8 objects:

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

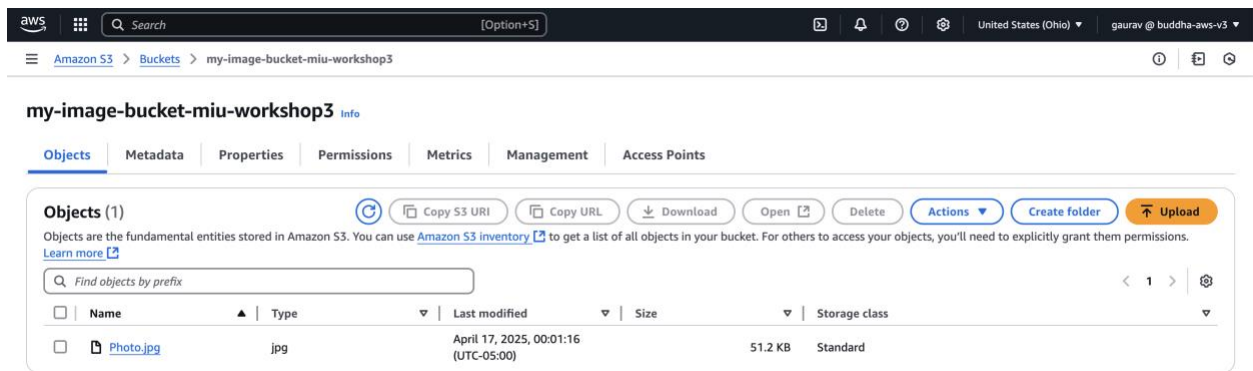
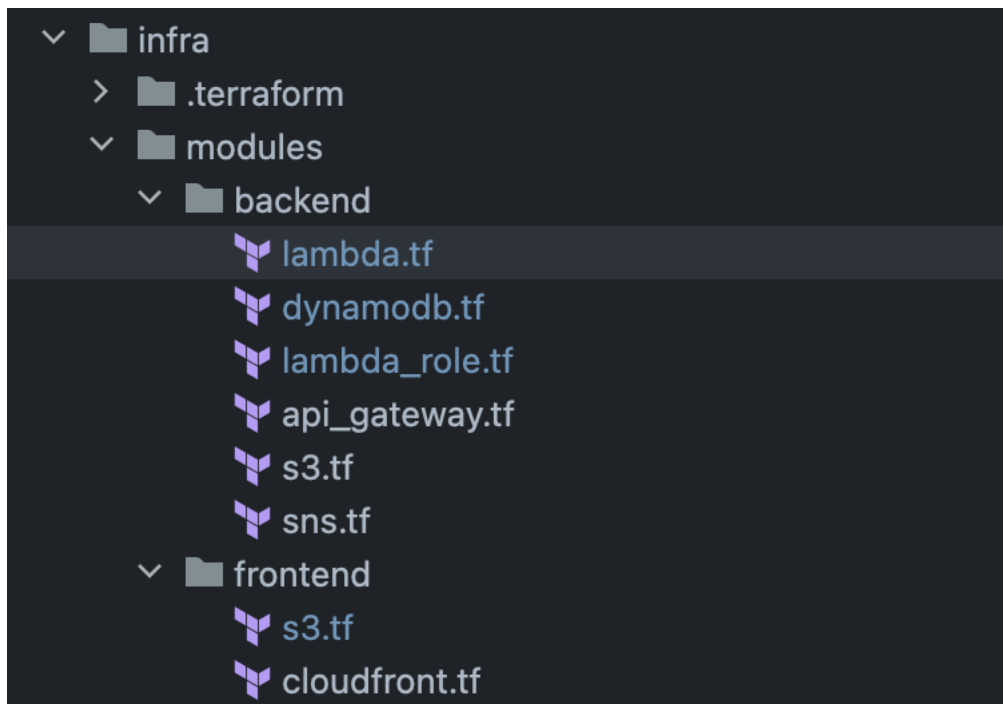


Image uploaded successfully.



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
  }
}
```