PROJECT REPORT – CE308

TITLE: SERVERLESS CLOUD RESUME

Ayesha Faheem	**2022131**
Fatima Shabbir	**2022173**
Khadija Nawaz	**2022250**
Lamia Asad Khan	**2022258**

SUBMITTED TO:

MA’AM SAFIA BALOCH

GHULAM ISHAQ KHAN INSTITUTE OF ENGINEERING SCIENCES AND TECHNOLOGY (GIKI)

Serverless Cloud Resume Project Report

Objective

To build a cloud-hosted personal resume website with:

- A visitor counter backed by DynamoDB.
- A contact form that stores messages in DynamoDB.
- Serverless architecture using AWS Lambda, API Gateway, S3, and CloudWatch.
- Hands-on practice with IAM, and CORS configurations.

Architecture Overview

This project is built using a fully **serverless architecture** on AWS. Here's the high-level design:

```
User → S3-hosted Website → API Gateway → Lambda → DynamoDB
                                           ↓
                                    CloudWatch Logs
```

Technology Stack:

Component	Service/Technology
Frontend Hosting	Amazon S3
Backend Logic	AWS Lambda
API Endpoints	Amazon API Gateway
Database	Amazon DynamoDB
Infrastructure (Optional)	AWS CloudFormation
Monitoring	Amazon CloudWatch Logs

Project Components:

1. Frontend (S3 Hosted Website)

Files:

- index.html: Contains resume layout, contact form, and visitor display
    - About Me
    - Visitor Counter
    - Contact Form
    - Skills
    - Experience
    - Projects
- style.css: Basic styling for layout
- script.js: Fetches visitor count and sends form data to backend

Key Features:

- Real-time visitor count using API
- Contact form to send messages directly to backend
- Deployed as a public static website using **S3 Bucket**

Deployment:

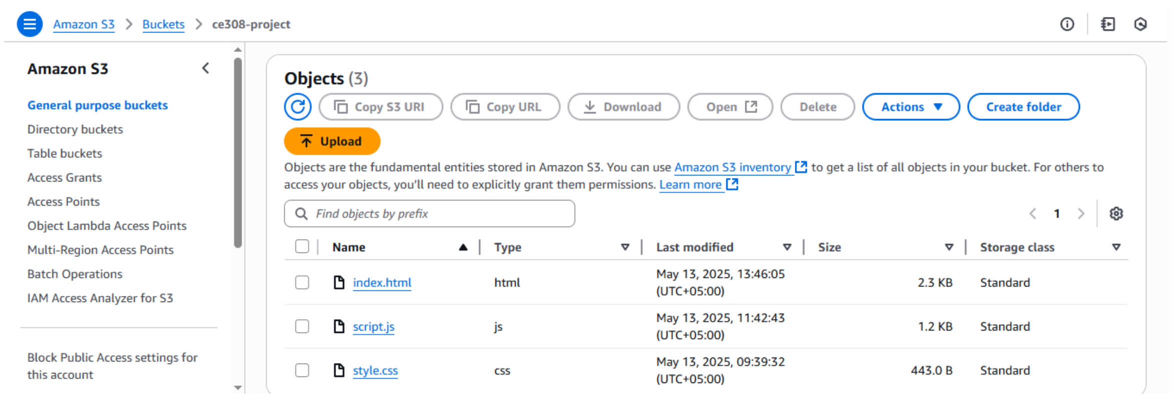
Created an S3 bucket

Enabled **Static Website Hosting**

Uploaded HTML, CSS, and JS files

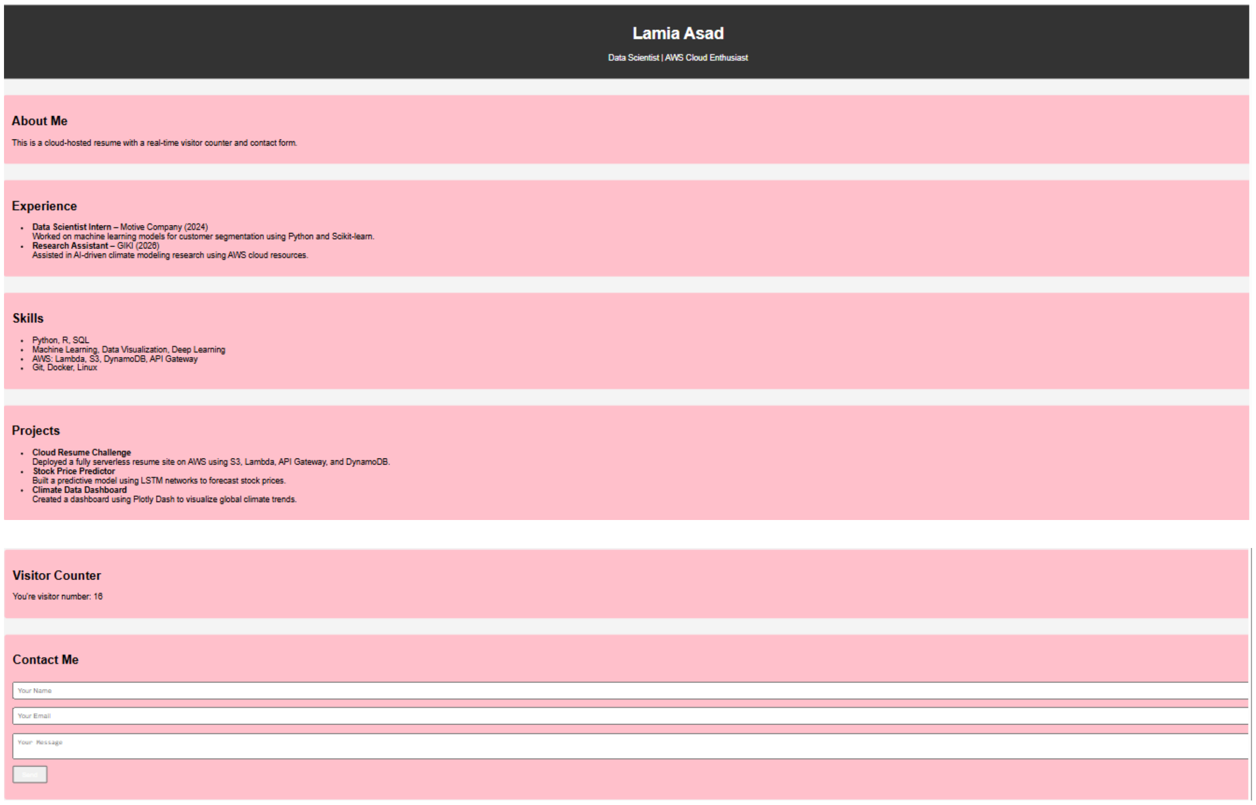
Configured **bucket policy** for public access

Website accessed via S3 Website endpoint



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

Name	Type	Last modified	Size	Storage class
index.html	html	May 13, 2025, 13:46:05 (UTC+05:00)	2.3 KB	Standard
script.js	js	May 13, 2025, 11:42:43 (UTC+05:00)	1.2 KB	Standard
style.css	css	May 13, 2025, 09:39:32 (UTC+05:00)	443.0 B	Standard



Lamia Asad

Data Scientist | AWS Cloud Enthusiast

About Me

This is a cloud-hosted resume with a real-time visitor counter and contact form.

Experience

- **Data Scientist Intern –** Motive Company (2024)
  Worked on machine learning models for customer segmentation using Python and Scikit-learn.
- **Research Assistant –** GIKI (2026)
  Assisted in AI-driven climate modeling research using AWS cloud resources.

Skills

- Python, R, SQL
- Machine Learning, Data Visualization, Deep Learning
- AWS: Lambda, S3, DynamoDB, API Gateway
- Git, Docker, Linux

Projects

- **Cloud Resume Challenge**
  Deployed a fully serverless resume site on AWS using S3, Lambda, API Gateway, and DynamoDB.
- **Stock Price Predictor**
  Built a predictive model using LSTM networks to forecast stock prices.
- **Climate Data Dashboard**
  Created a dashboard using Plotly Dash to visualize global climate trends.

Visitor Counter

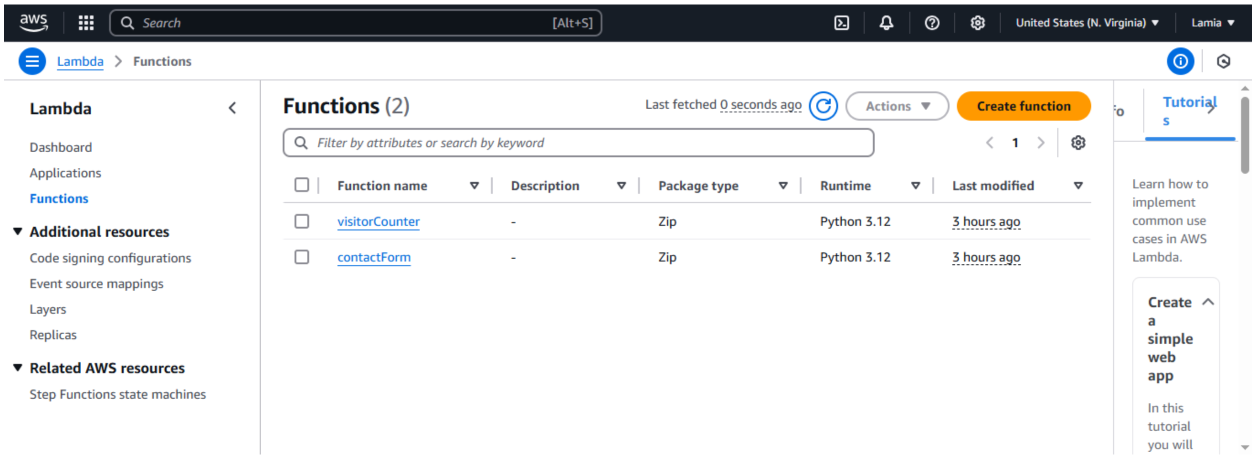
You're visitor number: 16

Contact Me

Your Name

Your Email

Your Message



aws
Search [Alt+S]
United States (N. Virginia)
Lamia
Lambda > Functions
Lambda
Dashboard
Applications
Functions
Additional resources
Code signing configurations
Event source mappings
Layers
Replicas
Related AWS resources
Step Functions state machines
Last fetched 0 seconds ago
Actions
Create function
Filter by attributes or search by keyword
Tutorials
Learn how to implement common use cases in AWS Lambda.
Create a simple web app
In this tutorial you will

Functions (2)

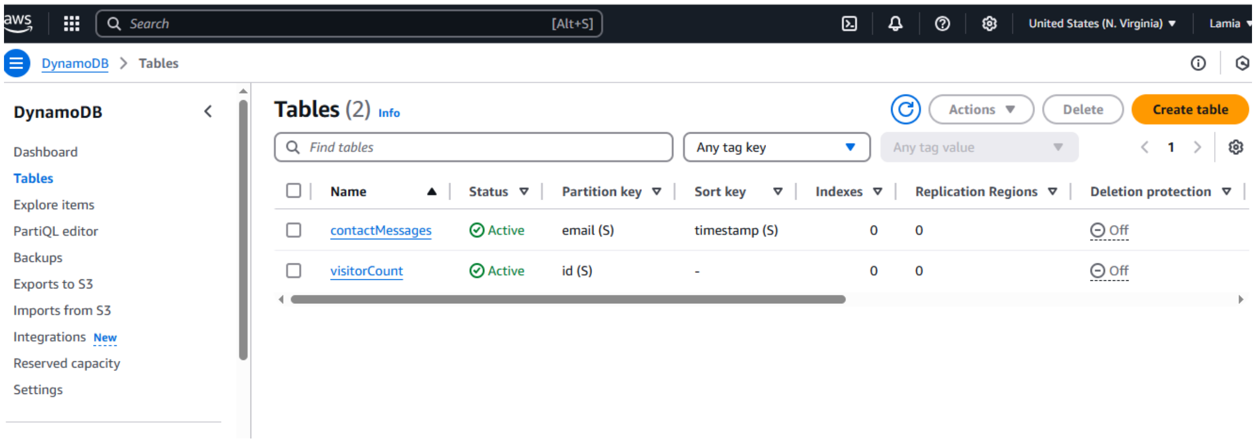
Function name	Description	Package type	Runtime	Last modified
visitorCounter	-	Zip	Python 3.12	3 hours ago
contactForm	-	Zip	Python 3.12	3 hours ago

2. Visitor Counter

Backend Logic:

- **Lambda Function**: visitor_counter.py
- **DynamoDB Table**: visitorCount

- Partition Key: id (String)
- Initial Item: { "id": "visits", "count": 0 }



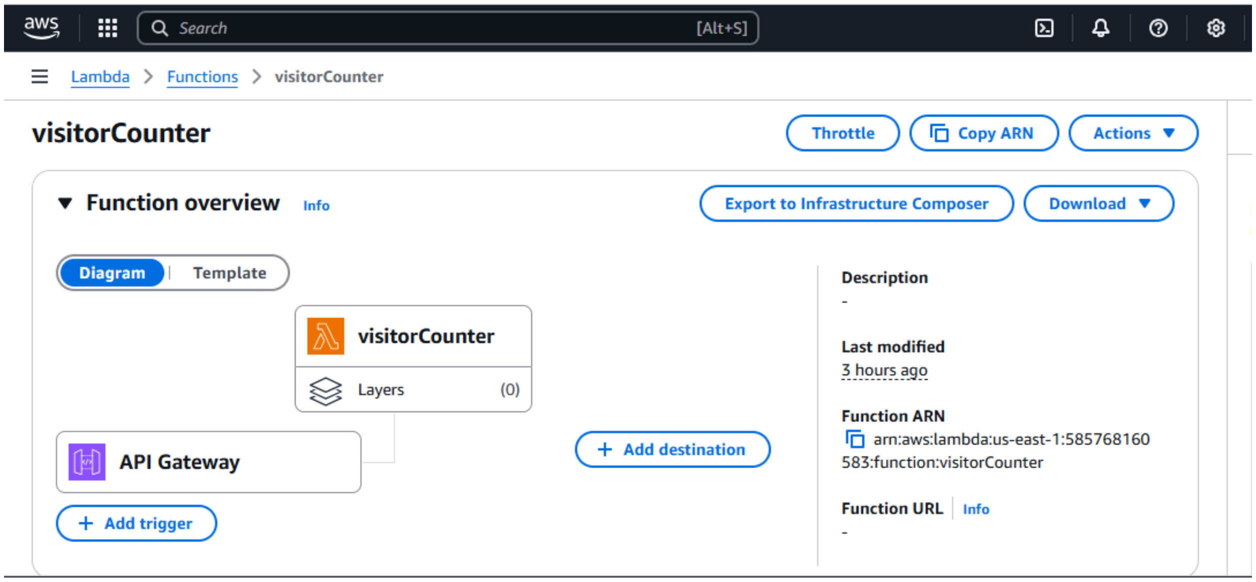
Name	Status	Partition key	Sort key	Indexes	Replication Regions	Deletion protection
contactMessages	Active	email (S)	timestamp (S)	0	0	Off
visitorCount	Active	id (S)	-	0	0	Off

Lambda Code Summary:

- Increments in the count value of 1 for every new visit
- Returns updated count to frontend

API Gateway:

- Method: **GET**
- Integrated with Lambda
- CORS Enabled
- Exposed via URL used in script.js



visitorCounter

Function overview Info

visitorCounter

Description
-

Last modified
3 hours ago

Function ARN
arn:aws:lambda:us-east-1:585768160583:function:visitorCounter

Function URL Info
-

3. Contact Form Handler

Backend Logic:

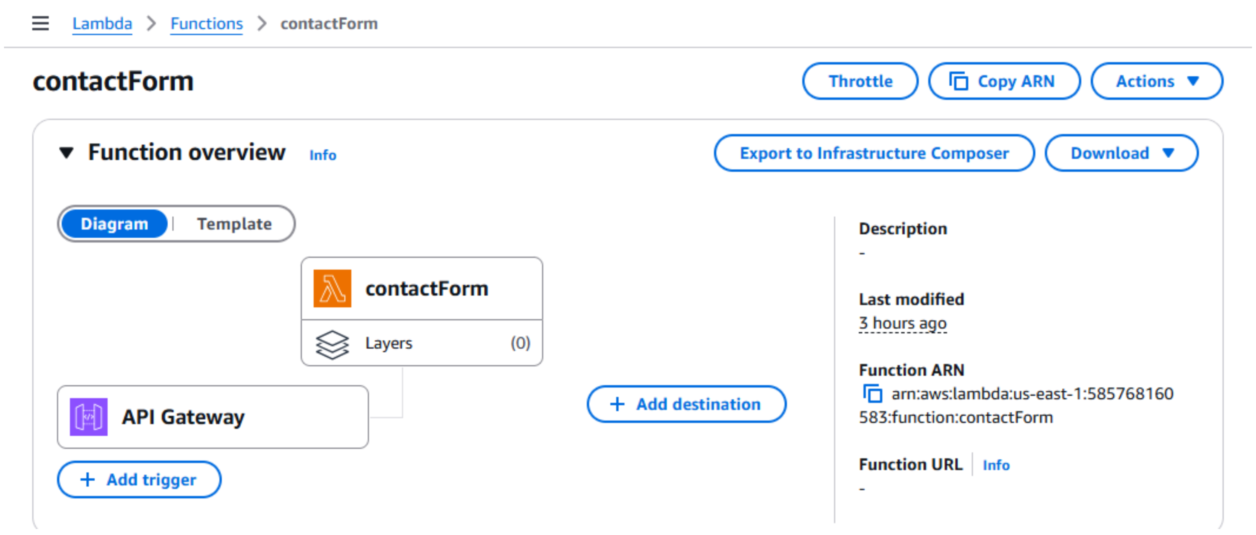
- **Lambda Function**: contact_form.py
- **DynamoDB Table**: contactMessages
  - Partition Key: email (String)
  - Sort Key: timestamp (String)

Lambda Code Summary:

- Receives from data (name, email, message)
- Stores each submission as a unique item using timestamp

API Gateway:

- Method: **POST**
- Integrated with Lambda
- CORS Enabled
- Exposed via URL used in script.js



contactForm

Function overview

contactForm

Description
-

Last modified
3 hours ago

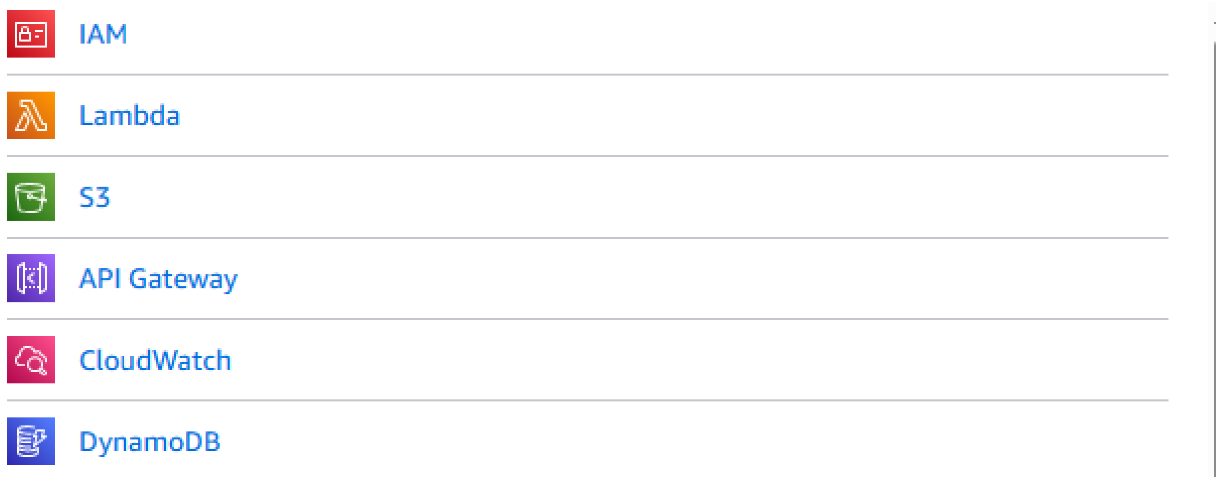
Function ARN
arn:aws:lambda:us-east-1:585768160583:function:contactForm

Function URL
-

4. Infrastructure Notes

While we manually deployed the services, this can be automated with CloudFormation using these templates:

- s3_bucket.yaml: For hosting
- dynamodb_tables.yaml: For visitorCount & contactMessages
- lambda_functions.yaml: For both Lambdas
- api_gateway.yaml: For API Gateway integration



IAM

Lambda

S3

API Gateway

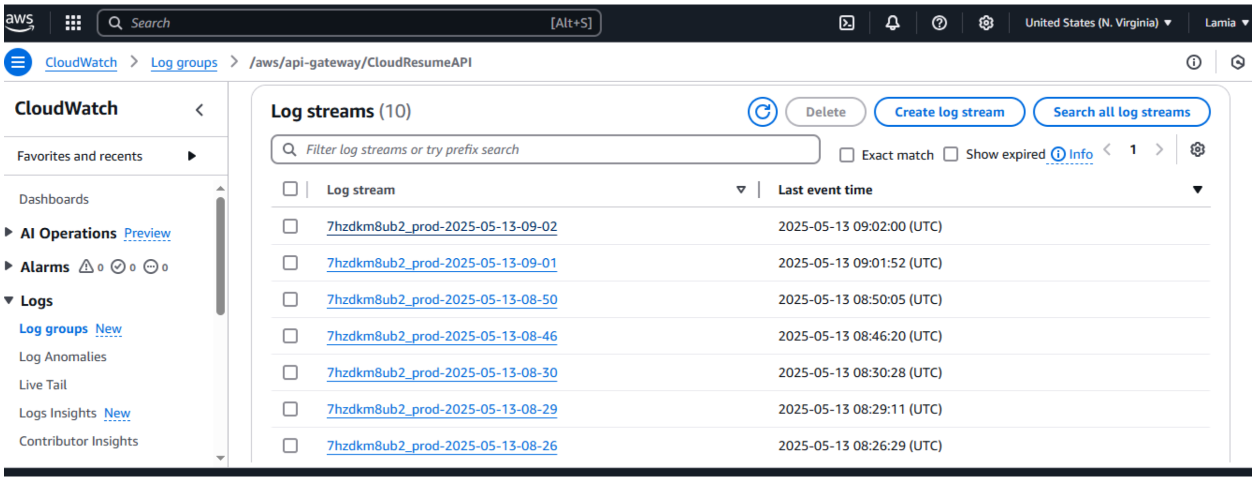
CloudWatch

DynamoDB

5. Monitoring & Security

- CloudWatch logs automatically enabled for Lambda functions

- IAM Roles created for Lambda with permissions to access DynamoDB
- API Gateway secured using CORS policy
- No hardcoded secrets or environment variables



Log stream	Last event time
7hzdkm8ub2_prod-2025-05-13-09-02	2025-05-13 09:02:00 (UTC)
7hzdkm8ub2_prod-2025-05-13-09-01	2025-05-13 09:01:52 (UTC)
7hzdkm8ub2_prod-2025-05-13-08-50	2025-05-13 08:50:05 (UTC)
7hzdkm8ub2_prod-2025-05-13-08-46	2025-05-13 08:46:20 (UTC)
7hzdkm8ub2_prod-2025-05-13-08-30	2025-05-13 08:30:28 (UTC)
7hzdkm8ub2_prod-2025-05-13-08-29	2025-05-13 08:29:11 (UTC)
7hzdkm8ub2_prod-2025-05-13-08-26	2025-05-13 08:26:29 (UTC)

6. Conclusion

This project shows proficiency in:

- AWS Cloud services (S3, Lambda, DynamoDB, API Gateway)
- Frontend development (HTML, JS)
- Cloud architecture design
- Serverless deployments and monitoring