NATIONAL UNIVERSITY
of Computer & Emerging Sciences

Project Statement: Application deployment on AWS infrastructure

1. In this project, you are required to deploy a cloud-based web application that follows the principles of modern cloud-native architecture.
2. You can either use your own project or fork a relevant project from GitHub (repository link must be submitted).
3. The application must consist of two separate components:
   - A backend service that performs CRUD operations, handles user authentication, manages a relational or NoSQL database, and supports secure file and image uploads.
   - A frontend client that consumes this backend via REST APIs.

Note: The project must be done in pairs, meaning no more than two people are allowed per group. Every instance, policy, security group, or deployment link must include your name. You will be evaluated on this, and failure to comply will result in a deduction of marks.

Project Functional Requirements

The application must fulfill all of the following functional criteria:

1. User Authentication
2. CRUD Operations (on at least one primary entity — e.g., Posts, Products, Tasks).
3. Database Operations: Implement user management and entity persistence using:
4. Amazon RDS (MySQL/PostgreSQL) OR
5. Amazon DynamoDB
6. File and Image Upload Support using AWS S3 Bucket.

AWS Cloud Infrastructure Requirements

You must deploy the system on the AWS cloud services, but in a production-like, secure architecture. All deployments must be public-facing with secured access controls:

1. Frontend (React/Angular/Vue/etc.)
   - Deploy on Elastic Beanstalk.

2. Backend (Node.js/Flask/Django/etc.):
    - Deploy on Amazon EC2 using a Docker container.
3. EC2 should be hosted inside a VPC with proper Security Groups and IAM roles.

4. Database Layer:
    - Use either Amazon RDS (PostgreSQL/MySQL) or DynamoDB.
5. File/Image Storage:
    - Upload and retrieve media assets using Amazon S3.
    - Set up S3 bucket policies for private and public access separation.
6. Security:
    - Implement and document:
    - IAM Roles and Policies for EC2, S3, and RDS access.
    - Security Groups for EC2 and RDS (open only necessary ports).
    - HTTPS access (use ACM if integrating Load Balancer optionally).
    - Least-privilege access model across components.

Bonus

Configure Route 53 for custom domain setup.

Security & IAM

You must provide:

- Screenshot evidence of IAM role creation and attachment.
- Configured inline or managed policies for services.
- Network access control (Security Groups and subnet access).

Submission Requirements

- GitHub Repo Link (Forked or Original).
- README with: Deployment Guide
- Live Demo URLs of frontend and backend (hosted on AWS).
- PDF Document with: Architecture diagram of AWS deployment
- IAM policy Screenshots
- Screenshots of AWS deployment and configurations

Important Notes

1. You are not allowed to use a full-stack framework that merges backend and frontend (e.g., Next.js full-stack apps).
2. Backend and frontend must be independently deployable.
3. All of the AWS services mentioned above have Free Tier but monitor billing closely to avoid charges.