Project Title:

Python-based Serverless User Registration System with S3 Image Upload and RDS Data Storage.

Objective:

To design and implement a serverless web application, developed in Python,

allowing users to register by providing their name, email, and profile picture. Upon

submission, the profile picture is uploaded to an S3 bucket, while user details (name

and email) are saved in an RDS MySQL database. An email notification of the

registration is then dispatched using SNS. The project ensures scalability, high

availability, and data security.

Background:

With AWS's comprehensive suite of serverless technologies, building scalable and

high-availability applications without server management has become a reality.

Python, with its vast ecosystem and the powerful Boto3 library, simplifies

interactions with AWS services, making it an ideal choice for crafting this solution.

AWS Secrets Manager and API Gateway bolster security, ensuring credentials are

safely managed and the endpoint remains secure.

Components and Functionalities:

AWS Lambda (Python):

● Central to the system, this Python application processes registration

requests.

● Manages the logic for saving user data to RDS and images to S3.

● Utilizes the Boto3 library for seamless integration with AWS services.

● Dispatches notifications through SNS upon successful registration.

Amazon S3:

● Stores profile pictures uploaded by users.

● Provides high durability and availability for images.

● Images are made accessible via unique URLs for retrieval.

Amazon RDS (MySQL):

● Serves as the primary data storage, recording user names and emails.

● Python's pymysql library facilitates database interactions.

● AWS Secrets Manager ensures secure database connection by

safeguarding credentials.

AWS Secrets Manager:

● Manages RDS access credentials securely.

● Directly integrates with the Lambda Python application to provide

secrets for database connection.

Amazon SNS:

● Notifies stakeholders (e.g., admin or user) of successful registrations

via email.

● The Python application, using Boto3, handles notification triggering.

Amazon API Gateway:

● Fronts the Python Lambda function, providing a secure and accessible

API endpoint.

● Handles incoming user registration requests and triggers the Lambda

function.

● Implements security features like rate limiting, CORS, and potential

integrations for user authentication.

Flow of Operations:

Users access the system's frontend, which could be a web page or mobile

app.

They fill out the registration form (name, email, profile picture) and submit.

The frontend sends the user data to the secured API Gateway endpoint.

The API Gateway activates the Python-based AWS Lambda function.

Lambda processes the data:

● Boto3 library aids in uploading the profile picture to S3.

● User details (name and email) are recorded in RDS using pymysql.

● An email is dispatched using SNS to notify relevant parties of the

registration.

The user receives a confirmation of their successful registration.