AWS Solutions Architect - Associate

Graduation Project Ideas

Author: Ayman Aly Mahmoud

ayman@manara.tech

Ayman Mahmoud | LinkedIn

Project Deliverables

Learners are expected to submit the following:

1. **Solution Architecture Diagram**
   - Create a visual representation of the solution architecture.
   - Tools such as Lucidchart or any other free diagramming tools may be used.
2. **GitHub Repository**
   - A public repository containing the complete project documentation (Please include the solution architecture diagram and the documentation in the README file).
   - Here is an example for structure and content guidelines.
3. **Optional Deliverable**
   - A live URL or a recorded video demonstrating the deployed solution on AWS (optional but encouraged).

Below are the details for 3 project ideas that learners can use for their graduation project.

Project 1: Scalable Web Application with ALB and Auto Scaling

Architecture: EC2-based

Description:

Deploy a simple web application on AWS using EC2 instances, ensuring high availability and scalability with **Elastic Load Balancing (ALB)** and **Auto Scaling Groups (ASG)**. The project demonstrates best practices for compute scalability, security, and cost optimization.

Key AWS Services Used:

- **EC2:** Launch instances for the web app.
- **Application Load Balancer (ALB):** Distributes traffic across multiple instances.
- **Auto Scaling Group (ASG):** Ensures instances scale based on demand.
- **Amazon RDS (Optional):** Backend database (MySQL/PostgreSQL) with Multi-AZ.
- **IAM:** Role-based access to instances.
- **CloudWatch & SNS:** Monitor performance and send alerts.

Learning Outcomes:

- Setting up **secure and scalable** EC2-based web applications.
- Implementing **high availability** using ALB and ASG.
- Optimizing **costs and performance** using Auto Scaling policies.

Project 2: Serverless Image Processing with S3 and Lambda

Architecture: Serverless

Description:

Create a **serverless image processing application** where users upload images to an S3 bucket, triggering an AWS Lambda function that processes and resizes the images before storing them in another S3 bucket.

Key AWS Services Used:

- **Amazon S3:** Stores original and processed images.
- **AWS Lambda:** Executes image processing (resize, watermarking).
- **Amazon API Gateway (Optional):** Expose an API for uploads.

- **Amazon DynamoDB (Optional):** Store metadata about uploaded images.
- **AWS Step Functions (Optional):** Handle complex workflows.

Learning Outcomes:

- Building **event-driven** architectures with Lambda and S3 triggers.
- Understanding **cost-efficient, auto-scaling** serverless applications.
- Enhancing security using **IAM roles and S3 bucket policies**.

Project 3: Serverless REST API with DynamoDB and API Gateway

Architecture: Serverless

Description:

Develop a **serverless REST API** using Amazon API Gateway, AWS Lambda, and DynamoDB to manage a simple to-do list or customer records. The API allows users to **create, read, update, and delete (CRUD)** data without managing servers.

Key AWS Services Used:

- **Amazon API Gateway:** Exposes REST endpoints.
- **AWS Lambda:** Handles API requests (CRUD operations).
- **Amazon DynamoDB:** NoSQL database for storing records.
- **AWS IAM:** Controls access via roles and permissions.
- **Amazon CloudWatch:** Logs and monitors API activity.
- **S3:** to host the front-end of your application

Learning Outcomes:

- Designing **scalable, event-driven** serverless applications.
- Implementing API Gateway with Lambda for **stateless execution**.
- Using **DynamoDB as a managed NoSQL database** with best practices.
- Securing APIs with **IAM roles and resource policies**.