CS 470 Project One Guidelines and Rubric

Competency

In this project, you will demonstrate your mastery of the following competencies:

- Apply cloud-based development principles and best practices in application development
- Develop applications that run on cloud-based frameworks

Scenario

You are a developer for **QnA**, a startup company. You just completed the development of a full stack web application and watched app usage take off after the implementation of in-house servers. It's an exciting time for the company. The company is growing, and you can feel the growing pains. The company is expecting a drastic increase in customers. You notice that the once-fluid office environment that allowed customers to choose their favorite devices and platforms has led to major limitations in collaboration. Development has become siloed, and sometimes only one key person can manage certain aspects of the application. To continue to allow your application to expand with the growing company, you need a solution for moving the application to a cloud-based framework.

In this project, you will explore cloud development frameworks for leveraging cloud infrastructure to support your application. This project will allow you to migrate your web application from the desktop to a cloud-based environment. All of these iterations will show you how the cloud is designed to make scaling up and growing your business more efficient. The IT team has determined that Amazon Web Services (AWS) provides a **Platform as a Service** capability that allows you to minimize system management and efficiently run services by dynamically scaling an application's capacity to run up and down. You will be leveraging the principle of elasticity.

Directions

Specifically, your web application must demonstrate your ability to do the following:

Frontend: Establish an S3 server to host a static Angular website.

- Prepare the client-side and web server setup, and apply the principles of serverless web hosting by moving frontend-Angular to the cloud.
- Prepare S3 storage buckets and objects and IAM security settings.

Now that you have created your containers, you can see there are benefits and pitfalls to this approach. Your director wants to ensure that your site is elastic, is secure, can process data quickly, and can keep up with the amount of traffic and transactions as the number of customers increases. Her thought is to utilize more AWS products and move the frontend to S3, a serverless solution.

You will be running your application in a serverless environment using a cloud-based database solution and detaching the Angular frontend and running it in the cloud, removing the need for the original server.

Backend and API: Migrate all of the server-side business logic.

- Demonstrate how Lambda containers can be used to distribute functionality in a serverless environment.
- Migrate all of the server-side business logic; apply principles of serverless API by writing code.
- Establish IAM security policy updates to protect the website and database.
- Change the API Gateway to connect the Lambdas.

As you explore native cloud services you will discover best practices that simplify the work of a developer, leaving them focused on developing rather than continuously working on maintaining the existing infrastructure.

In the final phase, you will deconstruct the full stack application APIs and turn them into microservices, making them fully cloud-native applications. This involves adding CRUD functionality and API Gateway using Lambdas to eliminate the need for the original MEAN server.

What to Submit

To complete this project, you must submit the following URLs to Brightspace:

- API URL / Endpoint: This is the backend of the project running.
- S3 URL / Endpoint: This is the project fully running.

Supporting Materials

The following resource(s) may help support your work on the project:

CS 470 Project One Guide

Specific instructions will be supplied in this guide.

Project One Rubric

Criteria	Exemplary	Proficient	Needs Improvement	Not Evident	Value
Serverless Web Hosting	Exceeds proficiency in an exceptionally clear, insightful, sophisticated, or creative manner (100%)	Prepares the client-side web server setup and applies the principles of serverless web hosting by moving frontend-Angular to the cloud (85%)	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include the omission of a URL, a screenshot of the server files, or the dist directory. Website must function fully and support CRUD capabilities (55%)	Does not attempt criterion (0%)	5
Lambda Compute Models	Exceeds proficiency in an exceptionally clear, insightful, sophisticated, or creative manner (100%)	Demonstrates Lambda compute models by using Lambda containers to distribute functionality in a serverless environment (85%)	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include illustrating how Lambda containers function or Lambda functions not reacting to events or properly handling CRUD actions (55%)	Does not attempt criterion (0%)	20
Serverless API	Exceeds proficiency in an exceptionally clear, insightful, sophisticated, or creative manner (100%)	Migrates all of the server-side business logic; applies principles of serverless API and demonstrates how they give the correct results when Lambda executes (85%)	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include one of these three examples when code executes on Lambda: 1) call the API, the connection from Lambda to DB, 2) Gateway to Lambda, 3) User to Gateway (55%)	Does not attempt criterion (0%)	15
Serverless Database	Exceeds proficiency in an exceptionally clear, insightful, sophisticated, or creative manner (100%)	Correctly creates a database with the appropriate structure that is able to perform the CRUD functions (85%)	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include an incorrect structure or the inability to perform any of the CRUD functions (55%)	Does not attempt criterion (0%)	20
CORS	N/A	Manually enables CORS as defined on the “CORS on AWS API Gateway” website (100%)	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include incorrect or partially complete CORS configuration (55%)	Does not attempt criterion (0%)	10
Create and Deploy API	N/A	Deploys the completed API using the ‘api’ deployment name (100%)	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include incorrect, partially complete, or not deployed APIs (55%)	Does not attempt criterion (0%)	10
Modify and Test Frontend	Exceeds proficiency in an exceptionally clear, insightful,	Applies changes to Loopback SDK configuration in the client;	Shows progress toward proficiency, but with errors or	Does not attempt criterion (0%)	10

	sophisticated, or creative manner (100%)	modifies environment products; builds and deploys the Angular app. The Angular application runs on AWS calling through Lambda to DynamoDB and successfully allows CRUD functions on Question and Answer (85%)	omissions; areas for improvement may include not changing the configuration of the URL to the AWS APIs, not changing the Loopback configuration to use query parameters instead of headers, failing to build the Angular application before copying to AWS S3, or failing to copy the right files to AWS S3 (55%)		
Cloud Security	N/A	Sets the permissions through policies and config to enable all of the other services with access to read-write and read-only (100%)	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include creating most or all of the roles and policies, but not configuring all of them correctly (55%)	Does not attempt criterion (0%)	10
				Total:	100%