

Nate Bennett

Cemal Tepe

Full Stack Development II

Southern New Hampshire University

27 October 2024

## CS 470 Final Reflection

<https://youtu.be/9cKjsoj5Hpk>

Hello, my name is Nate Bennett, and this presentation is about the intricacies of cloud development. Topics should include containerization, container orchestration, the serverless cloud, APIs, Lambda functions, databases, cloud-based development principles, and cloud security.

To migrate a full stack application to the cloud, it can be beneficial to use containerization to package applications and their dependencies to allow for consistency across different environments.

Docker is one of the popular tools used to containerize applications, and Docker Compose could potentially help with managing multiple container applications.

Docker Compose can simplify the deployment of potentially complex environments by defining services, networks, and volumes in one YAML file. This can help with replicating the same environment across development, testing, and production.

Serverless architectures allow developers to focus on code while the cloud provider manages infrastructure, scaling, and maintenance, potentially improving development efficiency.

Amazon S3 storage is highly scalable, generally accessible from anywhere with a device and an internet connection using APIs, and integrated with cloud services. Local storage, while generally swifter, is limited by the physical capacity of a device and can only be accessed locally.

Serverless APIs, such as those using Lambda functions with Amazon API Gateway, are scalable and allow for secure connections between services, helping with streamlining API requests.

Lambda APIs are endpoints for Lambda functions. Lambda functions could run functionalities like querying a database for data, and are accessed through Amazon API Gateway. They dynamically allocate resources based on demand, and scale with traffic.

MJS scripts are used to define what functionality Lambda functions should run.

The frontend and backend are integrated by connecting frontend API calls to backend Lambda functions using scripts.

MongoDB offers a flexible schema with a collections and documents structure, which could be helpful for complex queries and analytics. DynamoDB is optimized for high performance for applications with well-defined access patterns using a key-value pair structure.

Queries that were performed for this project include queries to get all stored question and answer data, as well as queries to get, create, update, or delete a question or answer on the database.

MJS scripts were produced for the GetSingleRecord, TableScan, UpsertAnswer, DeleteRecord, UpsertQuestion, and FindOneQuestion Lambda functions to run these queries.

Cloud platforms like AWS adjust resources by default based on demand, so that applications perform efficiently during high and low traffic times.

The pay-for-use model is set up so that only the resources that are used are to be charged to the AWS customer, helping optimize cost.

IAM roles and policies could be set up to limit access to data and functionalities, helping with preventing unauthorized access to these resources.

IAM roles define user permissions, and policies enforce what actions are allowed. So, by assigning roles to services like Lambda and API Gateway, access to these resources could be limited.

A role called LabRole was used to enforce security, the Lambda functions that were created were assigned this role, so, they have custom policies that allow only authorized users to access their functionalities.

Developers could secure the connection between Lambda functions and the API Gateway, Lambda functions and the database, and the S3 Bucket using IAM roles that limit access to only authorized users.

So, applications can be containerized to help them function consistently across different environments, cloud architectures are flexible, offer scaling by default, and have resource management so that application function efficiently during high and low traffic times, and IAM roles can be used to secure access to resources like Lambda functions.