

John Elbogen

CS 470

8/10/21

CS 470 Project Two Script

<https://youtu.be/LSMIoXTyt-Y>

1. CS 470 Project Two, Conference Presentation: Cloud Development
2. Hello, my name is John Elbogen a SNHU Student, Studying Computer Science with a focus in software engineering.
The purpose of this presentation is to examine and explain the process of moving an application to a serverless cloud environment including both front and backend development.
3. Containerization. AWS Cloud service provides multiple tools and servers for cloud migration. Containerization solves the problem of moving computing environments (ex. server to cloud based environment). Docker is a popular containerization application for building containers which hold application
4. Docker Compose allows for management of multiple containers. This application allows for multiple environment management allows for streamlining and continuous integration workflow. It is useful for rapid handling, testing, and development for multiple microservices Refer to image on slide
5. Serverless means you do not manage, or host dedicated hardware to run a server. This management is handled by the cloud provider using an event-driven setup. S3 storage is Amazon's cloud service providing storage for the internet, designed to make web-scale computing easier.
Some advantages of serverless include:
 - Easy scaling, horizontally and vertically
 - Cost reduction as the model is pay-per-use, vs local storage having a fixed cost regardless of use
 - Security is automatically managed saving production time (handled by provider)
6. **Serverless API Advantages**
 - Requests create and destroy servers in a 1-1 request fashion
 - Cost is based on number of requests which is efficient
 - Scaling is simplified and easier

Lambda API Logic.

- Lambda Proxy or non-proxy integration is used.
 - The input for the lambda function expressed as any combination of request headers, path variables, query string parameters, and body
 - Lambda will then execute the function code
- To deploy Lambda functions written Node.js, Java script code can be uploaded as a zip (plus dependent libraries)

Integrating the frontend with the backend.

- A request is sent to the API Gateway (Security handled here)
- The Gateway send the request to the correct lambda function

- Lambda function callback returns to API Gateway (Security handled again)
- The API Gateway returns to the browser

Explain Diagram

7. MongoDB uses JSON-like Documents while dynamoDB uses tables, items, and attributes
Some Queries from that I write
 - Insert Question/Answer
 - Find one question
 - Get one record
 - Delete one record

Javascript files are used for each Query and then executed as lambda functions

The database is stored as key-value pairs using DynamoDB

Explain Image

8. Elasticity
 - Dynamic Allocation
 - Meets Various Loads
 - Pay-for-use model
 - Based on Request
 - \$1.00 Per million

Explain Image
9. Securing A Cloud application is done with three main sections, Access, Policies, and API Security That work together.

- Bucket Access is managed through permissions in Amazon S3.
- This can be used to prevent any level of authorization the attached microservices

- An IAM role is an identity you can create that has specific permissions.
- A policy is a specific access activity for a role.
- Created Lambda Access to Question and Answer Table policy.

- Lambda Authorizers allow control over access to API methods using bearer token authentication
- Cross-origin resource sharing (CORS) lets you control how your REST API responds to cross-domain resource requests
- Permissions secure access to S3 Buckets

10. So what are the key takeaways?
 - Serverless Cloud Provides easier management, scalability, and security
 - Serverless migration from local is done through containerization (Docker), serverless API and lambda functions, and S3 Bucket services using Amazon AWS
 - Backend and Frontend security are managed through Amazon API and different security protocols such as bucket permissions, lambda authorizers, and IAM role policies

