Project Two Conference Presentation: Cloud Development

https://youtu.be/f5RrHBMWlUc

**Overview**

Hello, I am Gregory Greene. A Computer Science major at Southern New Hampshire University. During this presentation on cloud development, we will cover multiple topics including Containerization, Serverless Cloud Computing, API, Lambda, Databases, Cloud based development, and Security.

**Containerization**

Containerization is a process that bundles an application’s code, and all the files necessary to run into a package, called a container, for easier management. Docking is a common platform that supports containerization. It is used for cloud-based development and manages containers to simplify deployment and application management. The Lift and Shift Model is a practice that allows for easy migration to the cloud, as the entire application is pushed to the cloud without any changes being made to the code.

**Orchestration**

The use of Docker Compose simplifies the deployment of complex applications through the ability to manage multiple containers as a single stack. The services required can be defined in a YAML file making setup easier to manage. Docker Compose makes scaling, maintaining, and updating applications simpler for developers.

**The Serverless Cloud**

What is “serverless” and what are its advantages? During development the required infrastructure is managed by the cloud service provider. This reduces startup time and cost for developers. Necessary scaling is handled by the cloud service. What is S3 storage and how does it compare to local storage? Amazon S3 is a cloud-based object storage service offered by AWS. AWS provides security features to the developer to protect their data. Cloud based storage allows users to store and retrieve data from anywhere with a connection to the system.

**The Serverless Cloud (pt. 2)**

API & Lambda. Serverless API allows for faster development, less resource use, and cost-efficiency. Lambda API logic involves the process of creating a RESTful API through AWS Lambda functions that will manage and API requests and responses. We do this through producing scripts such as Lambda functions, API gateway configurations, and request & response scripts. In order to integrate the frontend & backend in AWS developers must define and test the API, connect the front/backend, and implement data rendering.

**The Serverless Cloud (pt. 3)**

Databases. What are the data-model differences between MongoDB and DynamoDB? The differences between these two include: Scalability, data structure, and querying. DynamoDB stores data based on key-values, while MongoDB uses document-oriented based storage. Common queries performed include Get, Post, Push, Table Scan, and Update/Delete Item. In order to do this, we must produce scripts including Lambda functions, API gateway configuration, and request/response scripts.

**Cloud-Based Development Principles**

Elasticity refers to how a cloud-based system can dynamically adapt to the number of resources being consumed. Systems can adjust to changes in traffic, amount of data being processed, or computing power required. This allows for a system to scale according to the workload and reduce wasted resources. The Pay-for-use model is a pricing model in which consumers only pay for the resources they use. Charges typically include storage, bandwidth, and computing power. This can be cost-efficient for smaller projects, as opposed to traditional fixed cost models.

**Securing Your Cloud Application**

Unauthorized access can be prevented through the use of authentication, defense in depth, and frequent security and system updates. Roles are created to define permissions needed by AWS to perform different functions. Policies are created and assigned to roles to specify the permission given to a specific role. Implementing roles and policies provides security through providing minimum required permissions for a role and reducing the risk of a role having unnecessary access to AWS resources. Connections between Lambda and Gateway and the S3 Bucket can be secured through AWS Identity and Access Management and API Gateway Authorization. Securing the database and AWS Lambda can be done through implementing Defense in Depth, including minimum required permissions, AWS Identity and Access Management, and frequent policy verification.

**Conclusion**

In conclusion, advantages of Cloud-Based Development include Scalability, Security, and a Pay-for-Use model. Scalability is managed by the cloud service provider and handles resource management, storage, and computing power based on traffic. Cloud service providers implement their own security measures to protect their client’s data. A good practice to follow when developing is Defense in Depth, implementing multiple layers of security including encryption, and access control. And with cloud development the Pay-for-Use model can help reduce startup cost for developers.