Chloe Britton - Project Two Conference Presentation: Cloud Development

Date: December 10, 2024

YouTube Video Link: https://www.youtube.com/watch?v=zVQm5wCAchE

My name is Chloe Britton. I'm getting my bachelor's degree in Computer Science and soon to be Full-Stack Developer.

The purpose of the PowerPoint is to articulate the intricacies of cloud development. Using containerization is

important when transferring your full-stack application to the cloud. The reason being that containers provide

a consistent and portable runtime environment, ensuring that your application runs the same way across development,

testing, and production environments.

The way you can use these containers easily is by using Docker Compose. Docker Compose allows you to define and

manage multi-container applications with a single YAML configuration file. This simplifies the orchestration of

containers, enabling you to start, stop, and scale your entire application stack with a single command which we

will see in the next slide. Docker Compose is a great tool when you're using multiple containers. Docker Compose

lets developers quickly use the entire application stacks with a single command, making sure there is consistency

across development and production environments. The tool also improves dependency

management by handling container

start-up order and making sure critical services are ready before others begin. Docker Compose versionable

configuration files make updates easy to track and roll back if needed, making it simple to share setups with others.

This efficient, portable approach makes Docker Compose important for any team deploying complex applications, which

you can see from this picture how portable everything is in its own little container able to run.

Serverless computing allows developers to focus on writing code without the need to manage servers, which means ease

of use, scalability, and security. Services like S3 simplify data management by getting rid of the need to predict

storage capacity, as it automatically scales for you. This makes S3 more efficient and flexible than traditional

local storage while integrating with serverless for easier and secure application development.

Using Lambda for serverless architecture offers cost efficiency, scalability, and fast deployment. Lambda charges

only for the compute time used, reducing infrastructure costs, and automatically scales to handle varying traffic,

ensuring optimal performance without manual intervention. The typical Lambda API logic starts with a user request,

which is routed through API Gateway to invoke the Lambda function for processing. After processing, Lambda returns

the response via API Gateway back to the user. To set up, first create the Lambda function, then configure the API

Gateway, enable CORS for frontend access, and finally, integrate the frontend to interact with the API Gateway. This
setup ensures a highly scalable, cost-effective, and efficient solution for deploying cloud-based applications.

The key differences between MongoDB and DynamoDB lie in their schema flexibility and querying capabilities. MongoDB
uses a document model, making it ideal for applications with complex data structures, while DynamoDB's key-value and
document model is fast, predictable key-based access. MongoDB is great at advanced querying, indexing, and flexibility
for complex queries. DynamoDB is good at simple high-speed operations for things that require consistency.

Cloud-based development uses things such as elasticity, the ability to automatically scale resources up or down
depending on what you need, ensuring good performance without overdoing it. The pay-for-use model allows businesses
to only pay for the resources they use, making it cost-efficient. Serverless stops the need for server management,
helping developers to focus on application logic while the cloud provider handles infrastructure, scaling, and maintenance.

Making sure your cloud application is secure means implementing access policies and API security measures. Access
policies define who can access stuff in the cloud, using tools like IAM making sure that users and services only have

the permissions they need. API security focuses on protecting your application's APIs from unauthorized access and attacks.

This makes sure that Lambda and S3 buckets are secure. The way they do this is by using authentication, as well as with

secure communication protocols like SSL/TLS.

Containerization, serverless cloud computing, and cloud-based development are key for building scalable and efficient

applications in the cloud. Containerization lets applications and their dependencies be packaged in isolated, portable

containers, making sure there is consistency across environments and helping deployment. Serverless cloud computing stops

the need to manage infrastructure, enabling developers to focus solely on code while the cloud provider handles scaling

and resources. These things are supported by cloud-based development principles like scalability, elasticity, cost

efficiency, and automation, ensuring applications can grow, minimize costs through a pay-for-use model, and streamline

development processes through automated workflows and infrastructure management.