CS 470 Project Two Conference Presentation

Tyra Faye Austria

04/26/2025

CS 470 Final Reflection

Overview

Hi, my name is Tyra Austria and this is my CS 470 Project Two presentation. So the purpose of this presentation is to articulate the intricacies of cloud development to both technical and non-technical audiences.

Watch my video here: https://youtu.be/9FbWacOOxmM

Containerization

Containerization means packing an app and everything it needs into a small, portable box called a container.

- Lift-and-shift: Move the app as-is into a container.
- Refactor: Tweak the app to fit better into a container.

Docker is the main tool for creating containers.

Rare info: Docker containers share the same operating system kernel, making them faster and smaller than VMs.

Store containers on Docker Hub or Amazon ECR.

Manage lots of containers with Docker Swarm (simple) or Kubernetes (for big apps).

Orchestration

Docker Compose simplifies the orchestration of multi-container applications by using a single YAML file to define and manage services. With one command, it can launch all containers consistently

across development and testing environments. This makes setup faster, reduces configuration errors, and streamlines workflows for small to mid-sized projects.

The Serverless Cloud - Serverless

Serverless computing lets you build and run applications without managing servers. It automatically scales and charges only for actual usage, making it cost-efficient and low-maintenance. AWS S3 complements this by offering scalable, reliable object storage. Unlike local storage, S3 is cloud-based, highly durable, and accessible from anywhere, making it ideal for modern applications.

The Serverless Cloud - API & Lambda

A serverless API using AWS Lambda removes the need to manage infrastructure while enabling scalable, event-driven logic. The Lambda function handles backend processes and is triggered via API Gateway. Scripts are written to define endpoints, process requests, and return JSON responses.

The Serverless Cloud - Database

MongoDB uses a flexible, document-based data model where data is stored in JSON-like documents with dynamic schemas. This allows for nesting and varied structures within a collection. DynamoDB, by contrast, uses a key-value and wide-column data model that requires a defined partition key (and optionally a sort key) for fast and predictable access patterns. It's optimized for scalability and performance rather than flexibility.

Cloud-Based Development Principles - Elasticity and Pay-for-use model

Elasticity means the cloud automatically adds or removes power based on how busy your app is, without you having to manage it. Some services, like AWS EC2 Spot Instances, even let you bid for extra servers at cheaper prices.

CS 470 Project Two Conference Presentation

With the pay-for-use model, you only pay for what you actually use - even down to the millisecond with services like AWS Lambda.

Securing Your Cloud App - Access

To prevent unauthorized access in our AWS environment, we followed the principle of least privilege. Only trusted users or services were given access, with multi-factor authentication enabled for user accounts. IAM roles and policies restricted access to only the necessary resources.

Securing Your Cloud App - Policies

In AWS, roles define what a service can assume, while policies define what actions are allowed. We created custom policies that granted specific permissions-like read/write access to S3 or database queries-ensuring that each role had only what it needed to function securely.

Securing Your Cloud App - API Security

We secured API connections using HTTPS through API Gateway. Lambda functions communicated with databases over encrypted channels, and S3 buckets were configured with access policies and server-side encryption. These layers protected data in transit and at rest.

Conclusion

Thank you for your time.

Serverless Architecture - Cloud development enables deploying applications without managing servers. Using services like AWS Lambda and API Gateway simplifies backend logic and scales automatically based on demand.

Cloud Storage and Data Modeling - Cloud platforms offer scalable storage solutions like Amazon S3 and databases like DynamoDB, which require efficient data modeling using partition keys for

performance, contrasting traditional databases like MongoDB.

Security and Access Control - Securing cloud applications involves IAM roles and custom policies to manage permissions, along with encrypted communications between services to protect data in transit and at rest.