Lindsey DeLorenzo

10/19/2025

CS470

Module 7

Project 2

https://youtu.be/cLIGlwrc8r8

Hello and thank you for joining me today. My name is Lindsey, and this is my CS 470 Module 7 presentation for Project 2. The purpose is to address cloud development.

**Containerization and Orchestration**

The first portion of this presentation is about Containerization and Orchestration. The containerization or bundling of application code, stores all files and dependencies into one package or container and is very useful for Cloud Services. For this Project, we were tasked with creating a Docker Compose with YAML service within Amazon Web Services and used API Gateway for the scripts, Lambda functions and for the security and user authorizations we used an IAM service.

Why would Docker Compose be used? It has everything needed for an application to run. It holds all the code and dependencies, runs it all, and tests it all. Making it easy to build, run, and deploy any application. It has a simple yet straightforward configurations for developers, it can be accessed portably to build or troubleshoot, and comes with AWS support features.

**Serverless Cloud**

Serverless systems are very useful in comparison to local servers. Serverless storage is when the application and data are stored remotely in a cloud unlike a physical local server tower. It is easier and cheaper to maintain cloud storage. Physical servers have to be cleaned, given electricity, kept in stable temperatures, and are large in size. Local storage is very expensive to start. They need to have large amounts of storage to hold all the application data and all future application updates.

With serverless data is making upgrading easier for growing companies. With S3 storage data is always protected and backed up automatically. Meaning any server crashes will be easily managed and accessible. Cost and maintenance are very straightforward with S3.

**API Advantages**

Lambda API has many advantages from its data being serverless, cost per usage, simplicity to create, easy to add security, and its handling of large traffic requests. API has 2 ways to handle the traffic through throttling and caching. It also uses access keys for API calls to help with keeping data secure.

**MongoDB vs DynamoDB**

MongoDB is a document-type database that allows for complex query requests. It is more flexible for more native datatypes, however, it can have data duplication and indexing issues.

DynamoDB is a key-focused database and has less complex queries. However, it fast, scalable, and serverless. DynamoDB being part of AWS has all the helpful perks of automatic back-ups, server management, security, and upgrades provided by AWS itself.

**Cloud-based Development Principles**

Elasticity and Pay-for-use go hand in hand. Elasticity will automatically adjust the resources that are being used. Pay-for-use Models are for only being charged for the data being used. The data being the resources stored, the application itself and the calls being requested when it runs. This pricing is upfront and has major cost saving benefits.

**Securing Cloud Applications**

Securing the cloud applications with S3 would be used through S3’s IAM roles and policies. IAM create roles with specific privileges and access. The role is who has the permission and the policy will have the selected service, resources, and permissions. A role with authentication would have more access than a public user. The policies we used have header mappings that control the response of the access methods.

**Conclusion**

Cloud development has more benefits for applications for any company. It allows for straightforward costs. It can be used from multiple locations. It can be easily developed between teams with its dependency support. There are strong security and authentication features. Back-up options are much safer for troubleshooting crashes.

Thank you so much for joining me in this presentation today and thank you for your time.

**Citations**

* Aleksic, Marko. (December 26, 2024) *What Is Docker Compose: Definition, Usage, Benefits.* PheonixNAP Global IT Services. <https://phoenixnap.com/kb/docker-compose>
* Afreen, Sana. (May 20, 2025). *What is Docker Compose: Example, Benefits and Basic Commands*. Simplilearn Resources DevOps. <https://www.simplilearn.com/tutorials/docker-tutorial/docker-compose>
* Villanueva, John Carl. (July 2, 2025). *Amazon s3 vs local storage - where should you store files uploaded to your file transfer server?* Jscape by Redwood. <https://www.jscape.com/blog/amazon-s3-vs-local-storage-where-should-you-store-files-uploaded-to-your-file-transfer-server>
* Jungco, Kezia. (February 15, 2023, Updated February 25, 2025). *Cloud Storage vs. Local Storage*. TA Technology Advice Blog. <https://technologyadvice.com/blog/information-technology/cloud-storage-vs-local-storage/>
* Slingerland, Cody. (February 11, 2025). *ECS Vs. ECS Vs. Lambda: The Ultimate AWS Comparison*. Cloud Zero Blog. <https://www.cloudzero.com/blog/ecs-vs-ec2/>
* Berkner, Erez. (September 16, 2019). *AWS Lambda vs EC2: Compared on Performance, Cost, Security, and More*. Lumigo Posts. <https://lumigo.io/blog/aws-lambda-vs-ec2/#:~:text=And%20the%20winner%20is:%20Lambda,to%20be%20added%20or%20deleted).>
* AWS. (date not given). Create you first Lambda Function. AWS Lambda Developer Guide. <https://docs.aws.amazon.com/lambda/latest/dg/getting-started.html>
* Adams, Shawn. (June 8, 2022) MongoDB vs DynamoDB Head-to-Head: Which Should You Choose? Medium Rockset Cloud. <https://medium.com/rocksetcloud/mongodb-vs-dynamodb-head-to-head-which-should-you-choose-a836655a2846>
* Bonisteel, Steve. (October 18, 2023) DynamoDB vs MongoDB: Choose One and Say No to SQL. Kinsta Blog. <https://kinsta.com/blog/dynamodb-vs-mongodb/>
* Amazon Web Services. (n.d.). *Introduction to AWS Identity and Access Management* (IAM). In AWS Identity and Access Management User Guide. Retrieved from <https://docs.aws.amazon.com/IAM/latest/UserGuide/introduction.html>
* Kovvuru, Ismail. (August 13, 2025) Elasticity, Scalability & High Availability in AWS Cloud Architecture. Medium Blog. <https://medium.com/@ismailkovvuru/elasticity-scalability-high-availability-in-aws-cloud-architecture-22dce7301271>
* Geeks for Geeks. (July 23, 2025) *Amazon Web Service - Introduction to API Gateway.* Geeks for Geeks DevOps. <https://www.geeksforgeeks.org/devops/amazon-web-service-introduction-to-api-gateway/>
* Solo. (September 25, 2025). *AWS API gateway The Ultimate Guide*. Solo Topics. <https://www.solo.io/topics/api-gateway/aws-api-gateway>
* Bonisteel, Steve. (October 18, 2023) DynamoDB vs MongoDB: Choose One and Say No to SQL. Kinsta Blog. <https://kinsta.com/blog/dynamodb-vs-mongodb/>
* Adams, Shawn. (June 8, 2022) MongoDB vs DynamoDB Head-to-Head: Which Should You Choose? Medium Rockset Cloud. <https://medium.com/rocksetcloud/mongodb-vs-dynamodb-head-to-head-which-should-you-choose-a836655a2846>
* AND YOU. Thank you for joining today.