Mehdi Mihir

CS-470

2/20/2025

Project Two Conference Presentation: Cloud Development

Video Link: Video

Presentation Narrative

Slide 1: Title Slide

Hello, my name is Mehdi. I really want to start off by sharing my experience going from a traditional full-stack application into a modern, more scalable serverless architecture. My journey isn't just about migration, but more about understanding how cloud-native principles can revolutionize software development and delivery.

Slide 2: About Me

Here's a little bit about me: I am a senior at SNHU majoring in Computer Science. My dad introduced me to technology at a very young age. After I graduate, I want to begin working as a Software Engineer or break into IT or Cyber Security.

My goal today is to break down the complex process of cloud migration into digestible insights. Whether you're a seasoned developer or just beginning to explore serverless technologies, this presentation will provide a practical roadmap for transitioning from traditional to cloud-native applications.

Slide 3: Initial Architecture Overview

Our starting point was a typical MEAN stack application. While powerful, it presented significant challenges in scalability and operational complexity. Each deployment required careful server configuration, and scaling meant manually provisioning new infrastructure - a time-consuming and expensive process.

Slide 4: Containerization Strategy

Our first transformation step was containerization using Docker. By encapsulating each application component in a container, we created a reproducible environment that could run consistently across different systems. Docker Compose became our orchestration tool, allowing us to define complex multi-container applications with just a few configuration lines.

Slide 5: Serverless Cloud Transition

Transitioning to AWS serverless wasn't just about changing hosting - it was a fundamental rethinking of our application architecture. We broke down our monolithic application into discrete, independently deployable functions. S3 replaced our traditional web server, Lambda functions replaced our Express routes, and DynamoDB offered a more flexible database solution.

Slide 6: Cloud Development Principles

One of the most transformative aspects of our migration was embracing cloud-native elasticity. Instead of maintaining constantly running servers, we now have an infrastructure that scales precisely to demand. Our compute costs dropped dramatically because we only pay for actual execution time.

Slide 7: Security Transformation

Security wasn't an afterthought but a foundational design principle. AWS IAM allowed us to implement extremely granular access controls. Each Lambda function, each API endpoint, and each database interaction is now secured with precise, auditable permissions.

Slide 8: Technical Implementation Highlights

By embracing serverless, we didn't just change our infrastructure - we transformed our entire development philosophy. Microservices became easier to develop, test, and deploy. Our continuous integration pipeline became more reliable and our ability to monitor application performance dramatically improved.

Slide 9: Lessons Learned

Our journey wasn't without challenges. We learned that successful cloud migration is incremental, requires continuous learning, and demands a shift in architectural thinking. The most important lesson? Serverless isn't just a technology - it's a strategic approach to building more responsive, efficient software.

Slide 10: Conclusion

As we look to the future, serverless and cloud-native architectures will continue to evolve. The key is to remain adaptable, curious, and always willing to challenge our existing architectural assumptions. Thank you.

Technical References

- AWS Serverless Architecture
- Docker Containerization
- Microservices Design Patterns
- Cloud-Native Development Principles

Contact

Mehdi Computer Science Student Southern New Hampshire University