Fathi Amran

4/28/2024

CS 470 Final Reflection

Video Presentation: https://www.youtube.com/watch?v=f3_8awvL8hk

CS 470 Final Reflection

Experiences and Strengths

CS470 wasn't just theoretical! It provided a deep dive into AWS cloud computing, giving me practical experience with in-demand technologies. I learned how containerization with Docker allows for clean application deployment and easy scaling across environments. Additionally, the course exposed me to the world of serverless solutions through AWS Lambda. By building and deploying code that runs on Lambda functions in response to events, I gained firsthand experience in this cost-effective and scalable approach. This hands-on learning has significantly expanded my skillset beyond the basics. Now, I can contribute to the development of modern web applications that leverage the power of cloud technologies, making me a strong candidate for jobs in today's dynamic tech industry.

My strengths as a software developer lie in my well-rounded skillset and commitment to quality. I'm comfortable working on both the front-end and back-end of web applications. This allows me to understand the entire development lifecycle and ensure seamless integration between different components. I possess a strong understanding of API design principles and best practices. I can effectively build and document APIs that are not only functional but also secure and easy for other developers to integrate with. Following industry best practices for security is important to me. I'm committed to writing clean and secure code, staying updated on potential vulnerabilities, and implementing robust security measures to protect applications from threats. I'm experienced in identifying and resolving bugs in code. I have a methodical approach to debugging, utilizing various tools and techniques to pinpoint issues and implement effective solutions. This ensures the applications I work on are reliable and function as intended. As a quick learner and effective team player, I'm eager to contribute to a team and keep growing as a developer. This makes me well-prepared for an entry-level full-stack developer position.

Planning for Growth

Microservices and serverless architectures can significantly improve a web application's future scalability and manageability. Breaking down an application into independent microservices helps scale individual functionalities based on traffic. Serverless functions on AWS Lambda

eliminate server provisioning and scale automatically to help keep costs in check for event-driven tasks.

To ensure robust error handling, leverage AWS CloudWatch Logs for centralized logging across all services. Implement clear error codes and informative user messages for graceful degradation. Regularly analyze these logs to proactively identify and fix recurring issues.

Cost prediction is easier with serverless compared to containerization. With serverless, you only pay for the resources your application uses. Containerized applications, however, require managing server instances, leading to potentially underutilized resources and unpredictable costs.

For future growth, microservices offer independent scaling and faster feature rollouts but require more development and testing effort. Serverless provides effortless scaling and cost-effectiveness but comes with vendor lock-in, potential cold start delays, and debugging challenges. The ultimate decision is based on application complexity, traffic patterns, and development resources.

Elasticity in both microservices and serverless allows an application to automatically scale up during growth spurts to handle increased traffic without performance issues. Additionally, pay-for-service models help keep costs in check as the user base expands.