

CS 470 Final Reflection

Debynyhan Banks

CS 470 Full-Stack II

Southern New Hampshire University

Professor Hodgman

June 25, 2024

<https://youtu.be/DKpBDVKydf0>

How This Course Will Help Me Reach My Professional Goals:

This course has equipped me with the essential skills and confidence needed to pursue my AWS Cloud Practitioner and AWS Developer certifications. Through hands-on experience with full stack web application development in the cloud, I have gained a deep understanding of cloud service concepts, containerization using Docker, serverless computing with AWS Lambda, API Gateway, and API security.

Strengths as a Software Developer:

My determination drives me to overcome challenges and continuously improve my skills. When faced with difficult problems, I persist until I find a solution, ensuring that my work is thorough and reliable.

Types of Roles I Am Prepared to Assume in a New Job:

Cloud Developer: With my extensive knowledge of AWS services and hands-on experience in deploying full stack web applications in the cloud, I am well-prepared to take on the role of a Cloud Developer. I can design, develop, and maintain cloud-based applications, ensuring they are scalable, secure, and efficient.

Types of Roles I Am Prepared to Assume in a New Job:

I am prepared to take on roles such as Cloud Developer, DevOps Engineer, Full Stack Developer, Solutions Architect, and Security Engineer.

How would you handle scale and error handling?

To handle scale and error handling, I would implement auto-scaling groups, load balancing, and use monitoring tools like AWS CloudWatch to detect and respond to issues promptly.

How would you predict the cost?

To predict the cost of using microservices and serverless architectures, I would utilize AWS's cost management tools such as AWS Cost Explorer and the AWS Pricing Calculator.

What is more cost predictable, containers or serverless?

Serverless architectures are generally more cost predictable than containers.

Pros and Cons for Expansion:

Pros:

- **Scalability:** Both microservices and serverless architectures can scale independently, allowing for efficient handling of increased loads.
- **Cost Efficiency:** Serverless architectures provide a pay-as-you-go model, ensuring you only pay for what you use.
- **Flexibility:** Microservices allow for flexible development and deployment, enabling different teams to work on various components simultaneously.
- **Resilience:** Decoupling services means that a failure in one component does not necessarily affect the entire application, enhancing overall system reliability.

Cons:

- **Complexity:** Managing a microservices architecture can be complex, requiring sophisticated orchestration and monitoring tools.
- **Latency:** The increased number of network calls between microservices can introduce latency.
- **Vendor Lock-In:** Relying heavily on a specific cloud provider's serverless solutions can lead to vendor lock-in, making it difficult to switch providers in the future.
- **Security:** Ensuring security across multiple services and endpoints can be more challenging compared to a monolithic architecture.

Roles of Elasticity and Pay-for-Service in Decision Making:

Elasticity allows the application to automatically scale resources up or down based on demand.

When planning for future growth, choosing an architecture that supports elasticity is crucial to handle variable loads without manual intervention.

Pay-for-Service:

The pay-for-service model ensures that you are billed only for the actual usage of resources, making it easier to predict and control costs. This model is particularly advantageous for startups and growing businesses as it allows for cost-effective scaling.

References:

Serverless Computing – Amazon Web Services. (n.d.). Amazon Web Services, Inc.

<https://aws.amazon.com/serverless/>