James Soto

12-11-2023

CS-470-Full-Stack-Development-II

SNHU CS-470 Full Stack Development II

CS 470 Final Reflection

What skills have you learned, developed, or mastered in this course to help you become a more marketable candidate in your career field?

The skills I learned in this course we how to integrate a local application into the cloud using AWS services. We took a what we learned in Full Stack Development I, utilizing the frameworks to build the cloud architecture. Prior to this course I developed my code on local platforms. This course taught me all the foundational frameworks that were needed to move into a cloud architecture which included containerization, the multiple cloud migration approaches, docker and docker compose, serverless application development, Amazon S3 storage, APIs and Lambda using AWS services, DynamoDB, and elasticity. The website that was ultimately created is : http://snhu-s3staticwebsite.s3-website-us-east-1.amazonaws.com/

Describe your strengths as a software developer.

The strengths that I possess as a software developer are that I am an extremely good problem solver. Another strength is that I am a leader and I have excellent communication skills in multiple languages that include Spanish, English, and Portuguese. I consider myself to be a quick learner, while I might not have the answer, solution, or experience at the beginning, I will adapt quickly and learn whatever is necessary or expected from me.

Identify the types of roles you are prepared to assume in a new job.

The roles that I am prepared for are software developer, full stack development, web development, using one of the languages I am most comfortable with, C++, Python, Java, and JavaScript.

Identify various ways that microservices or serverless may be used to produce efficiencies of management and scale in your web application in the future. Consider the following:

How would you handle scale and error handling?

AWS Lambda automatically scales to meet demand, eliminates the need for manual scaling, and automatically handles errors through retries and dead letter queues. A microservices architecture is a type of application architecture where the application is developed as a collection of services. It provides the framework to develop, deploy, and maintain microservices architecture diagrams and services independently. Each microservice is independently scalable. When traffic increases for a specific service, you can scale it independently without affecting others.

How would you predict the cost?

Serverless bills per execution, making cost prediction relatively straightforward. AWS provides detailed billing reports and cost forecasting tools. Estimating costs can be complex due to the distributed nature of microservices. Tools like AWS Cost Management can help monitor and predict costs based on usage.

What is more cost predictable, containers or serverless?

Serverless eliminates infrastructure management, resulting in only paying for actual usage. Cost predictability is high since you use the pay for use model. Containers require provisioning and managing infrastructure, leading to potential underutilization, and wasted resources. Cost predictability is moderate.

Explain several pros and cons that would be deciding factors in plans for expansion.

AWS serverless has a reduced overhead on the infrastructure management. Serverless is cost effective due to the pay per use model, and it is also elastic and very scalable. The cons of using AWS serverless is that your application is tied to AWS. Another con would be the limited infrastructure control and the increased complexity of debugging.

Microservices is agile, scalable and has fast development cycles. The cons are the increased complexity of development and infrastructure management while containing a higher overhead on the operational management.

What roles do elasticity and pay-for-service play in decision making for planned future growth?

Both microservices and serverless offer elastic scaling, allowing applications to automatically adapt to changing demand and optimize resource utilization. Both microservices and serverless use pay-per-use models, removing the need for up-front infrastructure costs and only paying for actual resource consumption. Microservices and AWS serverless offer many advantages in terms of management efficiency and scalability. The choice between them depends on the specific application requirements, team expertise, budget, and scalability needs.

YouTube Video of my CS 470 Presentation for Cloud Development

https://www.youtube.com/watch?v=HwBbejQkF9Y