Ron Bal

CS-470 Module 8 Final Reflection

2/25/24

Presentation link: https://www.youtube.com/watch?v=jwiCrfBzKGI

- **Experiences and Strengths:** Explain how this course will help you in reaching your professional goals.

This course has given me knowledge needed to create and migrate a cloud-based solution to AWS. These skills will be great if I find myself in a career of creating websites, creating or migrating to a cloud-based environment.

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

I wouldn't say I've mastered any skills in this course. It has, however, given me a blueprint to follow to further my knowledge with managing a full stack application through AWS.

- Describe your strengths as a software developer.

I don't feel I have any real strong skills as a software developer, but I have the tools necessary to lead me in the right direction and be successful as a software developer or engineer.

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

I think I would do well in an environment where I'm creating full stack applications through AWS. I really enjoyed the process and I'm excited to dive deeper and learn more of what AWS has to offer.

- **Planning for Growth:** Synthesize the knowledge you have gathered about cloud services.
  - 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?

  With handling scale and error each of the microservices can be scaled independently, creating a higher level of control over resources. Issues can be minimized and isolate the overall impact on a system.

  In a serverless solution, scaling is automatically handled by the cloud provider based on the incoming requests. Error handling is built in having automatic entries and a fallback mechanism.

- How would you predict the cost?

  Predicting costs in a microservice can be difficult as it involves forecasting usage for each of the services. It can be made easier with the use of monitoring tools to find and record patterns.

  Serverless is much easier as you pay for only the resources you use. A cost estimation tool is also offered, which can provide insight into what you can expect to pay.

- What is more cost predictable, containers or serverless?

  Containers are more predictable since you're only paying for the resources, but you may end up paying for low usage resources which could increase costs.

  Serverless you're only paying for the resources used which is cost effective, especially for workloads with varying usage.

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

Microservices are scalable, flexible, diverse, and can isolate faults. They are quite complex and have potential higher operational and communication overheads as well as high latency between services.

Serverless solutions automatically scale the project, have a pay-per-use model, and foster rapid development. However, you're locked into a vendor with limited control over the infrastructure and could have difficulty trying to manage your applications.

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

Both serverless and microservices scale resources elastically based on the demand needed for them. This allows resources to be utilized efficiently and reduce cost overheads.

The pay-for-service model is utilized by serverless architecture where you pay for only the resources that you used, encouraging efficient resources and cost optimization.