Nicholle Caudy

2/23/26

CS 470 Final Reflection

https://youtu.be/H6JDzl5dWkY

Experiences and Strengths

Throughout this course I have learned how to containerize and orchestrate an application using Docker and Docker Compose. I used Docker in a previous class, but this gave me a more in-depth understanding and expanded my experience. I also learned how to move an application to the cloud using AWS that included creating S3 buckets, APIs, Lambda functions, and IAM security features along with CORS. This has better prepared me for prospects in the future.

My strengths as a developer are attention to detail and troubleshooting. I am still learning and developing my skills every day, and I look forward to using all my newly developed skills in the future. The types of roles I am prepared to assume in a new job are data analysis and quality/testing I enjoy trying to interpret the information and trying to pinpoint where the problem may be and work to find a solution.

Planning for Growth

Microservices architecture divides an application into separate, independently deployable services that interact via APIs. Each service can be scaled on its own, allowing for more precise control over the infrastructure. Microservices do however require more effort in managing infrastructure, deployment, and orchestration. Microservices have a fixed infrastructure cost no matter what the usage. This may make the cost more predictable but may not be the most cost-effective option. Serverless architecture removes the tasks of infrastructure management and allows the developers to focus solely on writing and deploying code. Serverless architecture offers autoscaling and a pay-per-use pricing model making is a cost-effective option, especially with applications that may have an unpredictable workload. However, you can combine microservices and serverless which is known as serverless microservices allowing you to leverage the strengths of each architecture.

When evaluating expansion strategies, it is important to consider both the advantages and disadvantages. The first step, however, should be research to determine if expansion is a feasible option. When evaluating expansion, it is important to consider several potential drawbacks, including increased capital expenditure, the time required to scale infrastructure,

the costs associated with hiring and training additional staff, and the possibility of needing additional office space. Several advantages may be associated with expansion, including increased revenue potential, enhanced opportunities to collaborate with highly skilled professionals, and greater visibility and utilization of the application. Elasticity enables the cloud system to automatically adjust their resources to match changing workloads, scaling up or down based on current needs for your application which plays a role in expansion. Pay-for-service plays a role in planned future growth because you only pay for what you use. You can use the AWS pricing calculator to create an estimate for the cost of your use case and better help you decide if expansion is the right choice.