Bryan Rivera

10/21/2022

CS-470 Final Reflection

https://youtu.be/CQlgnWvB9GI

The skills I have learned and developed in this course that will help me become a more marketable candidate in my career field are Docker compose, Docker containerization and orchestration, MongoDB image and testing, Amazon S3, Lambda, API gateway, creating and using DynamoDB, and IAM roles and policies. My strengths as a software developer are problem solving, persistence, and adaptability. The types of roles that I am prepared to assume in a new job are software developer, software engineer, software tester, full stack software engineer, in reality any role because even if I don't know much about a role I will quickly learn and am able to adapt. Various ways that microservices or serverless may be used to produce efficiencies of management and scale in my web application in the future is that with scale handling the benefit of serverless is that it automatically scales based on what I need therefore I won't have to worry about scaling. With error handling I will be able to implement IAM roles and policies to be able to handle that and therefore I won't have to worry about that either because the IAM will have the roles and policies I need. I would predict the cost based on the incoming use of the web application because it only charges by the use of the website and not the

downtime when no one is using it with is also beneficial. The more cost predictable between containers or serverless would be containers. This is because with serverless the server only runs when an event triggers it, therefore you only pay for the time the server is in use and with containers you have to pay for the server usage even if it is not in use. Pros that would be deciding factors in plans for expansion would be the use of serverless because it manages the storage on its own and is able to allow your app to be elastic. A con with serverless is that because it become inactive when not in use there will be a latency between the requestion and the action.