

Experiences and Strengths

This course has helped me pull together everything I have learned. I was not familiar with containerization and AWS. This class has given me training on those aspects as well. I think having exposure in AWS and also migration to cloud makes me more marketable in my career field.

Problem solving is one of my strengths, which I believe will be very helpful in my career as a software developer. Additionally, I am a pharmacist. So having a domain knowledge will go long ways in my career in healthcare software developer. Additionally, I pay attention to details. That too will be a characteristic that will help me achieve my career goals. I am prepared to assume a role of full stack developer in healthcare quality improvement field.

Planning for Growth

Microservices creates distributed system of application which makes it portable. Creating microservices on the serverless environment allows multiple developers to work on different parts of the application at the same time. This allows for faster completion of projects. For error handling we can use AWS Step function to create workflow that can handle errors.

There are many factors to consider in planning for expansion. Capital expense required to store and deploy the application is very important. Having on-premise server requires allocation of resources and forecasting amount of computing resources needed. Availability of internet is also very important factor in making the choice. Organization also need to consider if the latency that is associated with serverless is acceptable for the given application.

In serverless model, capacity increases as needed resulting in no waste of resources. This allows the organization to focus on application development and do not have to reserve resources far in advance allowing for future growth.

In order to predict the cost of a project, we need to understand amount of resources needed for completing the work including any dependencies. Resources would include securing cloud resources that are autoscaling and developers. Serverless, usually, would be more predictable for the cost. With serverless, pay-as-you-go structure allows for not incurring any cost until computing resources are used.