Zachary Nicholas

December 17, 2024

CS 470 Full Stack Development 2

CS 470 Final Reflection

YouTube Presentation

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

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

During this course I have learned how to use docker and the AWS platform, through this I have learned how to create and run my own docker containers and link them together in order to make a Full stack application.

- Describe your strengths as a software developer.

I think my strengths are my ability to learn new concepts quickly and understand something as long as I am able to learn it at my own pace, my attention to detail is good, and my ability to work with a team and explain concepts to them about my work.

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

I think some roles that I would be able to assume would involve software development or engineering, Developing a full stack app either locally or in the cloud.

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?

I think using monitoring or tracking apps could help for me to understand and prepare for any case in which one of my apps would have issues or scaling, Using a software like Pulseway or Netdata to help me manage such things like over usage or for case where an error has occurred, these two software could be invaluable as they allow for notification and real-time monitoring of systems and services.

- How would you predict the cost?

I would look at other similar models and compare how a new application or system would look based off of our needs, another good option is to give an estimate of what the services could possibly need and look to AWS or another like system that offer pay-as-you-go which could allow for a basic banket plan and only charges us extra based on the usage.

- What is more cost predictable, containers or serverless?

I think that containers would be more cost predictable as it allows for us to allocate a specific amount of resources which will always be a fixed cost unlike with a serverless approach which could vary wildly based on the usage. I think that depending on the use case of the application either one of these solutions could work just depending if we have a set amount of usage we are predicting or if our usage could change drastically. The Idea of elasticity and pay-for-service approach are a neat concept and allows for us to more efficiently use resources and reallocate them when needed which could be very important if we are running multiple apps either locally or in the cloud which could allow us to not widely overspec a machine based on theoretical traffic we could get and take resources from either another system or from the main host system if they are running in a docker or VM style of way.