Mark Holden

CS-470 Full Stack II

10/23/2022

CS 470 Final Reflection

https://www.youtube.com/watch?v=-SB-JIDMKb0

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?

Gaining additional familiarity with the configuration and use of different web services could be helpful in my career.

Describe your strengths as a software developer.

My strengths as a software engineer are that I think critically about requirements and do not simply accept them at face value. I follow the test-driven development principles and work autonomously taking the initiative to clean and refactor code as needed.

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

I will probably be promoted to Senior Full-Stack Software Engineer soon, and I am preparing to accept that responsibility by working on projects that would typically be done by senior engineers.

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?

Scaling and error handling are both extremely complex topics and the answer to how to handle them is always situation dependent.

Within the narrow scope of a web-hosted microservice architecture, in general, scaling should be handled in two ways – scaling up and scaling out. If the bottleneck for the system is simply the number of requests that are being processed is greater than the system can handle, then scaling out – meaning just running more instances of the same service is a good solution to handle the requests. Any hosting service worth using should provide the ability to automatically scale out given a set of conditions. In a situation where requests are processed slowly because the hardware on which the application runs is underpowered, then scaling up can assist with the issue.

Error handling should be done by the application. If an error can be thrown, it should be handled. One of the ways to so that is by specifying, in the case of an API, what error codes can be returned by the endpoint. For example, a GET endpoint may return 200 OK, 401 Unauthorized, 403 Forbidden, and 404 Not Found, while a POST endpoint may return all the above plus 400 Bad Request. Calling functions need to handle success or failure responses and should at a minimum be logging failures to enable troubleshooting if the error was unexpected.

How would you predict the cost?

Cost predictions can be estimated by either selecting services or hosting platforms that will accommodate the system that have fixed prices and just adding up the prices for the tiers of service required or can be estimated for applications such as Azure Functions by guessing a rough order of magnitude of how many calls the service will receive (or basing this on current analytics, of available) and multiplying that by the cost per call.

It would be wise to include scaling in that calculation to ensure that as more users are added to the platform, the company becomes more profitable, not less.

What is more cost predictable, containers or serverless?

The cost of containers would be easier to predict on a fixed cost monthly basis. The cost of serverless compute would be easier to predict based on end user actions that would be billable for businesses that are set up to bill customers by usage, and it would also preserve the ratio between cost and revenue generated.

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

Pros

Expansion generally means more revenue.

Additional revenue from expansion can provide the means with which additional engineers can be hired or infrastructure can be purchased.

Cons

Expansion also means more costs associated with doing business.

A large expansion is probably not the time to switch from on-premises infrastructure to cloud infrastructure or vice versa.

If a switch was planned to go from on-premises infrastructure to cloud infrastructure or vice versa, doing it after a large expansion will certainly be harder than doing it before.

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

If costs can be deferred until when revenue is available to bear those costs, then that makes planning for future growth easy and desirable. One caution about that is that revenue needs to grow faster than costs to continue to be profitable. That sounds obvious, but if it is not carefully and intentionally planned, customer pricing could be set up in a disastrous way for the company.