Jared Meineke

10/23/2024

CS 470 Final Reflection

Video Presentation

Over the course of this class, we have learned about several important developer tools, like Docker and Docker Compose, as well as several AWS and cloud development features. Regardless of what specific career path I choose to follow, having a wider range of skills and knowledge of different technologies improves my flexibility as a developer and helps me to learn new skills and technologies even faster. I would say this is my strongest skill as a developer, I have often been described as a fast learner in previous courses and projects, sometimes working with professional developers. Along with this, I am thorough and methodical in my problem solving and I can adapt quickly to different coding standards for varying languages or companies. Because of these skills and my current experience, I feel comfortable fulfilling a wide range of entry level positions, as long as I'm given the opportunity to learn and grow within the position.

As far as planning for the growth of this application, AWS and cloud services lend themselves naturally to scaling and flexibility. Developing in a microservice architecture has the benefit of being able to seamlessly add new features, often without needing to affect any of the existing code. For predicting and handling errors, I would be sure to thoroughly test each piece of code I added to the service and be sure to add handlers to situations that often create errors, such as code that handles user input. As far as predicting the cost of the web service, some of the more important considerations come from usage, such as traffic, storage of files and in databases,

and in the computing power and time of functions. Serverless is more predictable in this regard, though, than more traditional models like local storage or containers. These often require a large upfront cost for hardware like servers. The scaling price model lets developers start up a service without needing as much initial investment, additionally providing a safety net in case a service does not meet its expected profits.

Likely the most important thing to consider when scaling up a cloud service is the increased profit versus the costs. It also creates additional security concerns and costs in development time and testing. The purpose of expanding is likely going to be to increase traffic, which will cause the cost for capacity to increase. Developers also need to be careful about tech debt, making sure that the added features do not introduce needless complexity. Some of the pros, however, include increased profits, a wider range of users, the opportunity to enhance the performance, reliability, and flexibility of the website. The elasticity and pay-for-service models also lend themselves to scaling a cloud service, allowing the cost and capacity to dynamically change based on the service's usage. This takes away considerations that might otherwise hamper a service's ability to grow, like the previously mentioned server capacity. Overall, cloud computing offers many benefits in developing and growing an application.