

Robert Golden

CS470 Final Reflection

6/26/2024

CS470 Final Reflection

<https://youtu.be/HchpKHRZzRw>

Experiences and Strengths

I learned how to transition an application to the cloud, implement an AWS gateway API, and create security policies including IAM roles. To be honest I don't know yet what my strengths are as a developer. I guess that I haven't found my niche yet. I am prepared to enter any lower level developer position, as I don't have a problem coding and I don't have any practical experience doing the job. Really the entire degree process was more to gain something that couldn't be held against me anymore. I still feel as though I should have gained a massive amount more of coding experience rather than the ability to fill out forms and documents, but this is the last semester and I'll figure the rest out there in the real world.

Planning for Growth

Handling scaling and error would be a challenging task. The serverless model automatically handles the scaling based on the load. I would handle errors with built in retries, and monitor function performance to find where bottle necks and errors are being produced. Using the serverless architecture and after some time I would be able to see trends in use. Since

this model uses pay-per-use, I can use the projected growth of usage and then I would be able to calculate the cost. Containers tend to be a bit more predictable, but with the ability to analyze usage trends cost shouldn't be too far off the prediction.

Here are some Pro's and Con's

Microservices	Serverless
<p>Pro's</p> <ul style="list-style-type: none">• Independent scalability• Flexibility• Fault isolation	<p>Pro's</p> <ul style="list-style-type: none">• Automatic Scaling• Pay-per-use means you only pay for what you use.• Simplified Operations
<p>Con's</p> <ul style="list-style-type: none">• Can be complex when managing multiple services across multiple services• Higher resource usage• Higher initial costs for infrastructure	<p>Con's</p> <ul style="list-style-type: none">• Cold Start Latency• Limited Execution Times for functions• Heavy reliance on platforms infrastructure•

Considering the roles of elasticity and pay-for-service, organizations can make informed decisions about their architecture to support planned future growth. The choice between microservices and serverless will depend on the specific needs of the application, workload patterns, cost considerations, and the desired level of operational complexity.