

Jonathan Boeglin 12/17/23

CS 470 Final Reflection

Project Video: https://youtu.be/59A_cTmrDeI

Experiences and Strengths

Within this course, I have learned how to properly setup and migrate a project consisting of a front and backend to a cloud service. Also, the process of containerization has been explained. Having the knowledge on how to complete these processes can be invaluable for companies looking to migrate, maintain, or modify their systems with regards to cloud-based computing. With these skills, I believe my list of strengths as a software developer include full stack development, secure coding, and reverse engineering. This knowledge will allow me to assume roles in a new career such as a programmer, DevOps/DevSecOps tech, or even a project lead with my management skills learned from my past degrees.

Planning for Growth

Planning for growth in terms of cloud-based content is an important part of successful cloud integration. One part of this process is handling scaling and errors. Scale can be predicted by knowing your company's needs, and their plans for expansion. If the service will be utilizing a much larger amount of data by a certain time, slowly scaling up through elasticity and adding necessary parts over time can reduce how quickly costs increase. For error handling, dedicating part of the scaled resources to recognizing and dealing with errors and uploading it with the relevant parts that it covers can prevent gaps in the detection system.

Another area of planning is cost prediction. While some cloud services offer pay-for-service, it is still important to know what it will cost to run with the current system and resources, as well as running with updated scaling and traffic. Again, this prediction can be

known by what the plans for scaling of a pay-for-service system are, and slowly increasing the load so that the cost doesn't skyrocket all at once. In addition, it is important to keep in mind that overall, containers are more cost predictive than serverless options. The best usage for containers are for dedicated systems that run for long periods of time. The best usage for serverless is for small applications that are used less regularly.

Pros and cons to consider for expansion would include if the space gained is going to be completely utilized, if any wasted space is worth the cost, if expansion will improve the service and not bloat the system, and if the current method is worth continuing or if a more cost-effective alternative exists. Elasticity and pay-for-service can help in this decision by allowing for more fluid expansion with only adding space as needed and paying for only what is used. This can cut down on costs and allow for expansions that may normally have to wait for more storage necessities or better cost-effectiveness.