Ren Welck

SNHU

CS-470

10/24/2024

Final Reflection

https://youtu.be/mwM6rlHIlZY

In reflecting on my time in CS-470 and its impact on my skills as a developer I can see how this course strengthened my grasp of cloud service concepts, my technical skills and my adaptability. The skills I have gained are the ability to know how to develop and deploy a full-stack web application in the cloud. I also learned how to design efficient APIs, that ensure thorough testing, which are essential in the real-world development environment. I also learned about cloud services' ability to be scalable, automated, and security management.

This also challenged my problem-solving skills and my ability to adapt to new technologies. I am now prepared to take on roles that would require a mix of development and cloud infrastructure knowledge, such as DevOps engineer, or cloud architect, or similar roles. With this experience in cloud-based app development, I am ready to take on roles involving full-stack development, API design, and cloud deployments, I am also equipped to support growing companies that require stability and scalability.

When planning for the growth of a cloud-based web application, it is crucial to consider efficient scaling, reliable error handling, and cost management. Starting with microservice and serverless architecture can simplify scaling and reduce operational costs, as an example

serverless functions can be triggered as needed which would be cost effective for varying loads. To handle scale, I would use auto-scaling policies for serverless functions, enabling elasticity based on traffic. Error handling would be managed through monitoring tools like AWS CloudWatch for automated alerts and rapid response to performance issues.

Between containers and serverless options, serverless is often more cost effective for unpredictable loads, as costs are tied to usage. However, containers might be more predictable for applications with a steady user base, as reserved instances can lower expenses. The pros of serverless would be cost effective, scales automatically, and reduces the need for server management. The cons are it is potentially higher cost for stable applications with predictable loads. Containers' pros are there is more control over infrastructure, stable costs with reserved instances, and cons are they require more maintenance and management.

These factors influence my planning for growth by ensuring resources are used only as needed, preventing over provisioning. Elasticity allows the application to handle surges in demand, while the pay for service model keeps costs manageable, as resources are dynamically adjusted based on usage.

In conclusion, CS-470 has provided me with a comprehensive foundation in cloud development and future proofing applications, empowering me to make strategic decisions for scaling and optimizing web applications.