

Jennifer Wells
8/25/24
CS 470 Final Reflection
<https://youtu.be/XvYBe1wNKM4>

This course has provided me with a deep understanding of cloud-native development, particularly with AWS services. As the industry moves towards cloud-first strategies, having the ability to design, deploy, and manage applications in the cloud is crucial. This course has equipped me with the skills to build scalable, secure, and efficient applications using modern cloud technologies, making me a more competitive candidate in the job market.

I have learned how to design and implement cloud-native architectures using AWS services like Lambda, S3, and API Gateway. Mastery of serverless computing has enabled me to build applications that scale automatically with demand, reducing operational overhead. By learning Docker, I've gained the ability to package applications into containers, ensuring consistency across different environments. Understanding AWS IAM and security best practices has enhanced my ability to secure cloud applications against unauthorized access. These skills position me as a more marketable candidate, particularly for roles that require expertise in cloud technologies and modern application development practices.

My ability to quickly learn and adapt to new technologies has enabled me to stay current in a rapidly evolving industry. I have strong communication and teamwork skills, essential for collaborating with cross-functional teams in software development projects.

With my knowledge of cloud architecture and services, I can design scalable and secure cloud solutions for organizations. With experience in both front-end and back-end development, I am prepared to take on full-stack roles that require a comprehensive understanding of web application development.

Jennifer Wells

8/25/24

CS 470 Final Reflection

<https://youtu.be/XvYBe1wNKM4>

Throughout this course, I've gained insight into how cloud services can transform traditional software development and deployment practices. I understand the importance of leveraging cloud-native services to create flexible, resilient, and scalable applications. The course has also highlighted the benefits of microservices and serverless architecture in building modular and maintainable systems.

By breaking down applications into smaller, independent services, I can scale individual components based on demand, improving resource utilization. Error handling can be isolated to specific services, reducing the impact of failures. By breaking down applications into smaller, independent services, I can scale individual components based on demand, improving resource utilization. Error handling can be isolated to specific services, reducing the impact of failures. Serverless functions like AWS Lambda automatically scale based on demand, and with built-in error handling, failures are managed more effectively without manual intervention.

Costs are predictable based on resource usage for each service, but managing multiple services can lead to higher operational costs. Costs are more granular and easier to predict as you only pay for the actual execution time of functions. However, unexpected spikes in usage can lead to unanticipated costs. Containers offer more predictable costs, as you can manage the resources allocated to them. However, scaling containers requires managing infrastructure, which can add complexity and cost. Serverless computing offers more fine-grained cost control but can become less predictable if usage varies significantly.

Using microservices, each service can be scaled independently. It allows for the use of different technologies best suited for each service, and failures in one

Jennifer Wells

8/25/24

CS 470 Final Reflection

<https://youtu.be/XvYBe1wNKM4>

service doesn't affect the entire application. However, managing multiple services can be challenging.

In serverless, there is no server management which reduces operational overhead, it seamlessly handles traffic spikes, and you pay only for what you use. However, initial invocation of functions may have a delay.

With elasticity, the ability to automatically adjust resources in response to demand ensures that the application can handle varying workloads without over-provisioning, which is crucial for managing costs during growth. With Pay for Service, this model aligns expenses with actual usage, allowing for cost-effective scaling. However, it requires careful monitoring to prevent unexpected costs during periods of high demand.

This course has provided me with the skills and knowledge necessary to transition into roles that leverage cloud technologies. By understanding how to effectively utilize microservices and serverless architectures, I am prepared to contribute to the development of scalable, efficient, and cost-effective web applications in the future.