

Monticia Dunn

CS-470 Module Eight

8/15/2023

[Presentation Link](#)

### **CS-470 Module Eight: Final Reflection**

This course was very helpful in helping me pursue my professional goals. I'd like to become a well-rounded software engineer that's not only proficient in programming but also able to handle common DevOps practices, like setting up containers with Docker and deploying an application to the cloud. Now I can say I have those skills and experience with these popular technologies when I'm applying for software engineering positions. I think my strengths as a software developer include my problem-solving skills, resilience, and resourcefulness. I'm prepared to assume any role on a software development team whether that's programming the application, testing the project, project management duties as well as DevOps procedures.

Both the microservices architecture and serverless architecture make management and scale more efficient in web applications, they produce those efficiencies in different ways however, for instance, when it comes to scale, serverless platforms will dynamically allocate resources based on incoming requests and so will microservices but the resources get allocated based on which microservices are receiving more traffic. Both architectures are also less error-prone since serverless platforms will have error-handling capabilities built-in that can be customized by the developer providing consistency in how errors like retry mechanisms are handled. With microservices the developer will have to set error-handling up themselves but due to separation of concerns, an error in one microservices is less likely to shut down the entire application. When it comes to cost, serverless platforms are more predictable due to the way a user gets billed based on usage of resources whereas with a microservice, the usage can vary across each microservice making it harder to track.

All in all, companies will have to weigh the increased cost of expansion with the potential increase in revenue and decide if the elasticity and pay-for-service models will provide the best of both worlds. This can be different for every project as not all applications will require the overhead of building a system based around being able to dynamically scale its resources up or down based on demand or being able to pre-provision resources to predict costs.