

Experience and Strengths:

When working on the assignments and projects of this course, I have both learned and developed the skills needed to work with containerization, Docker, AWS, and cloud integration. Containerization and Docker were new topics for me altogether, so learning about the architectural design, workings, and concepts will help when needing to work with simple concepts and systems in the future. Being that I aim to work as a software engineer and the technical world seems to be migrating more and more towards cloud development and environments, having worked with AWS and cloud integration will prove beneficial when searching for work in the computer science field. Additionally, I would say that some of my greater strengths are becoming self-sufficient in new technologies on a faster pace and time management to be able to get numerous assignments taken care of within a deadline and in a timely manner. In a new job, I could perform roles such as a software engineer, full-stack engineer, cloud application developer, and other related roles.

Planning and Growth

As proven with previous research into microservices and serverless designs, some of the ways that microservices and serverless designs might produce efficiencies for management and scalability would be through the pay-per-use models, reduced maintenance and cost on servers, and the ability to manage different roles and policies for different microservices and resources. Microservices then have analytics to go along with the pay-per-use models, which can help the developer determine the level of scalability that they would need to anticipate and determine the predicted costs. Equally so, these same analytics should be able to display when errors occur. Upon needing to deal with the error handling itself and maintenance of the system, developers should be able to make changes to the necessary files within the serverless cloud environment before redeployment thus allowing the web application to stay running as much as possible. Additionally, serverless designs would realistically be more cost predictable because it is a pay-per-use service that won't potentially remain idle and waste resources.

When planning for expansion, some factors that should be considered are the cost, storage capacity, speed, functionality, and elasticity. Cloud servers could end up being more expensive being that other people are maintaining the server that it is being held on, however if you don't necessarily know how big of a server you will need, cloud servers allow pay-per-use models to only charge you for the usage tiers that you would need. This means that if you have fluctuations or are unsure of your actual necessity, then you don't end up paying for too much or having not enough space causing problems. Also, with speed, there are typically regional options that tend to the server to be closer to a location reducing the length of time to connect with the server. As for functionality, if the application has heavy graphic intensive visuals, cloud servers might not be the right option due to capabilities.

Elasticity and pay-for-service both end up giving the developing company the ability to use more or less resources depending on its needs at the moment. This is beneficial for companies that don't know what to expect when coming out with new products, upgrades, or services that hopefully increase usage, but still be beneficial in case usage stays the same or drops.