

Apurva Shukla

20th August 2023

CS 470 Final Reflection

<https://youtu.be/X7G84zOYiVE>

The skills I have learned from this course includes using Docker to create a container that is able to run the front-end and back-end of a full-stack application. This was achieved using Docker Compose to create the containers in a predictable and makes it easy to replicate, then moving the application files to the cloud, namely AWS services. My strengths as a software developer lie in being able to understand how to manage services and anticipate solutions to problems that might arise in the future, along with the fact that I tend to find solutions that are cost-effective and efficient. After the learnings in this course, I believe I am fit to take on roles such as Developer, Tester, System Operator and DevOps.

This course has allowed me to understand the importance and significance of using serverless models, which provide for a relatively hassle-free experience when compared to legacy solutions. The course has taught me that error handling is not directly handled by the team developing the product, but by the serverless service providers. Error handling still has to be handled by the team developing the product, and therefore should use similar infrastructure so that when the applications are scaled, the errors are logged correctly. The cost of this application can be predicted by looking at the resources that will be potentially used by the application, the potential reach of the application in the market and the price of the service provider itself. In most cases, containers are more cost predictable as they are usually more responsive because they are always running, and also have a limited set of resources already

assigned to it, which means there is a base charge of using containers and only paying extra when scaled up. These are the pros of migrating to a cloud-based service model. The cons include added latency, lack of control over the infrastructure being used by the service provider - which may result in worse performance and also affected by outages, and the unpredictability of the costs should the application have an influx of traffic. When planning for future growth, businesses need to anticipate increases in demand. Elasticity allows them to scale resources as needed, ensuring that the infrastructure can handle higher workloads without compromising performance. Elasticity enables businesses to avoid overpaying for resources they don't need. They can scale down during periods of lower demand, saving costs.

Pay-for-service models allow businesses to allocate resources efficiently and avoid overcommitting financially. They only pay for what they use, making it easier to manage budgets during planned growth.