

# CS 470 Final Reflection

Justin Haby

Southern New Hampshire University

Cs-470 Full Stack Development 2

Prof. Nizar

08/19/2022

**YouTube Presentation Link:**

[https://youtu.be/C14w\\_XdPLZQ](https://youtu.be/C14w_XdPLZQ)

**Experience and strengths**

In the course CS 470, I explored the underlying detail development in the cloud. I discovered the tools necessary in cloud development to understand migrating a full stack application to the web. Tools such as docker, docker-compose, and Kubernetes were essential in understanding cloud computing architecture.

In migrating the full stack application to the cloud, I employed two necessary methods, which are rehosting and rearchitecting. In the rehost method of migration, no changes were applied to the code base of the full stack application. This implies the project was migrated 'as is. However, in the latter method, code changes occur to successfully migrate the full stack application to the cloud, which borrows directly from the rearchitect restructuring of the application's code base. In the usage of docker and docker-compose, some discoveries essential in the business were unveiled. For instance, when using multiple containers and services, the use of docker becomes problematic; therefore, docker-compose. This is because docker-compose indicated capabilities of handling various containers and services.

Moreover, the course explored the concept of serverless computing. A term that sounded a little misguiding at first, but it turns out to be a genius revolutionary idea of pay as you go. This idea entails a client only paying for cloud resources that are utilized compared to traditional computing that entails a fixed payment method by the cloud service vendor. I also increased my competence in using S3 storage, which is more efficient, highly available, and has a recovery protocol that is relatively steady compared to local storage.

Another skill I interacted with was lambda APIs and databases, which exposed me to scripts necessary for cloud development. To understand these concepts, I delved into the practice of writing the scripts and using them, which widened my knowledge of the cloud blueprint. As a software developer, I am a curious troubleshooter interested in the depth of systems and the applications of systems to the day-to-day business and social lives of earth's inhabitants. Therefore, I could assume roles at a job including software engineer, cloud architect, and cloud database administrator.

### **Planning for growth**

Peeping into the future and seeking relevant prospects for the days ahead, I would claim boldly that serverless computing and microservices are revolutionary inventions in the tech industry. Besides improving the cost of accessing cloud services, serverless computing would revolutionize the uses of the cloud as different clients would only purchase resources in the cloud, they are interested in using.

This will also balance a load of traffic on the servers of the cloud vendors as different servers would serve a reduced number of clients compared to traditionally when the servers had to handle the load all by themselves. Microservices are a clean way, quite analogous to division of labor and specialization, that, together with the serverless aspect of the cloud, would increase the security and integrity of data, improve the portability of projects across all kinds of devices and also improve the risk against unauthorized access by the use of secret managers, encryptions such as strong password policies and two-step authentication.

Scaling can be automated so that changes made to the full stack application can be accommodated smoothly into the cloud, and errors can be detected through context managers in the full stack apps that can reveal leakages, syntactic errors, and migration errors. The serverless cost is predictable when one knows the number of resources in the cloud one intends to use. However, containers would be more predictable since they are often fixed in some way. That implies scaling, and there might be already defined methods to scale. However, with the serverless, the resources predicted might be underestimated or overestimated.

In expansion, for instance, it would be not easy to indicate clients' traffic on a website hosted in the cloud. It is pretty easy to underestimate the number of users expected. This is where elasticity and pay for services come through. Elasticity ensures that the cloud can accommodate the traffic and changes whatever the situation.