

Micheal White
28 June 2024
CS 470 Final Reflection
https://youtu.be/K35cz_xnTmc

This course taught me several skills and concepts about cloud computing which has become increasingly popular today. I learned how to create containers using Docker for each component of my MEAN stack. I then used Docker Compose to create YAML files for the frontend and backend to get both up-and-running. I tested this by adding a new question to the database displayed through my Angular frontend website via localhost and checking that the question was added to the database.

After creating Docker containers, I started creating S3 containers in AWS to begin to move the application to the cloud. To set up my frontend S3 bucket, I uploaded all the dist directory files and assets into the bucket. Then I configured the bucket for static web hosting and defined bucket access to public for read access. I created an API using API Gateway, testing the API with a Lambda function using the GET method, and deployed the API. I then created more Lambdas to add the functionality to make queries to my app in the cloud, including searching the database, adding a question or answer, and deleting records.

The skills I have learned in this course provide me with a strong foundation of in-demand skills that developers need to be competitive. My love of learning will push me to expand my cloud computing skillset. My tenacity drives me to continue looking for solutions when I face roadblocks. Attention to detail helps me focus on writing code accurately to build components more carefully and to identify errors more quickly. I look forward to beginning my career in computer science as a software engineer or systems administrator. I also would like to look into a career in artificial intelligence and machine learning.

Serverless cloud computing provide organizations with options to create applications modularly rather than one large entity. Creating multiple smaller, independent components allows teams to find errors more easily. However, teams will need to understand how their functions and containers interact with each other to accomplish this efficiently.

Predicting costs is a challenge every business face since it is difficult to always know the exact costs of operations. Serverless computing options like AWS use a pay-as-you-go model where companies pay for what they need when they need it. This elasticity in price could benefit low to moderate use apps, but high volume or widely fluctuating use cases can make budgeting difficult. Microservices on containers are relatively cheap to provision but come with more upfront costs. They need compute power and memory to execute 24/7. Teams will have to plan for how much hardware or devices to rent for the app to run as intended while planning for future growth. Despite the higher cost, containers tend to be more cost predictable since the pay-as-you-go model can fluctuate.

Both architectures offer scalability, flexibility, and decentralization and are ideal for high volume requests applications. They allow developers to design and deploy code faster. To determine which is better, you need to consider your customers' needs and plans for future growth and features. You also need to think about the expertise of your team and the cost of upskilling and retraining. For our application which serves as a question and answer message board, serverless was the better choice for us. The app is gaining traction and the pay-as-you-go model is saving our company money. Yet, we have the flexibility to scale up as the app becomes more popular. We can use this time to upskill the team to prepare for future growth and adding new features.