

- **Experiences and Strengths**

- What skills have you learned, developed, or mastered in this course to help you become a more marketable candidate in your career field?
- Describe your strengths as a software developer.
- Identify the types of roles you are prepared to assume in a new job.

I have learned many new skills during this course. I have learned to use Docker to containerize an application. I have used Amazon Web Services (AWS), including Lambda for creating Application Programming Interfaces (APIs), Simple Storage Service (S3) buckets for storing database information, Using HTTP services such as PUSH, GET, PULL, and DELETE to create APIs that will manipulate database information. And learning how to create security features using Cross-Origin Resource Sharing (CORS) and Amazon's Identity Management (IAM). These skills will make me a likely candidate for a career in Full Stack Development.

Through other courses, I have also learned other computer science skills such as reading and developing code, using testing frameworks to test code, writing modular and commented code that is easy for someone to understand, and other skills that will help me to get a job in a computer science related field.

With my skills learned in this course, I should be able to get a job in full stack development or focus on either front-end or back-end services. Having mastered AWS services, I should be able to perform any job in this field.

- **Planning for growth**

- Identify various ways that microservices or serverless may be used to produce efficiencies of management and scale in your web application in the future.
- Explain several pros and cons that would be deciding factors in plans for expansion.
- What roles do elasticity and pay-for-service play in decision making for planned future growth?

Now that I know how to produce an application that focuses on microservices and being serverless, I can now use these skills to create applications that leverage those skills. By using them, I will be able to create applications that require less management, be of smaller scale, and be able to update or iterate parts quickly without having to take down the entire application. I can also update a database on the application with minimal interruption. Testing would be much easier as well, as the testing standards could be written in the application as a microservice, so the testing is performed each time an update is detected. Also, since the application would be based on Amazon's cloud services, it would be easier to determine how much the application would cost, since you only pay for what server time you actually use, instead of paying for idle time. As far as costs are concerned, it is easier to predict the cost of containers on cloud services, since their size and usage is more predictable. The serverless portion of the application would depend on how many people are using the application at any time, so it is less predictable.

Several pros and cons need to be considered when planning for expansion. How large is the application going to grow? How many users are expected to use it? How often will the

Rob Marlatt  
08/25/2024  
CS 470 Final Reflection  
<https://www.youtube.com/watch?v=a6IHr4U8usY>

application be used? Is it a local only application, or does it need to be available on the web?

These factors must be considered when determining whether or not to make an application serverless or not. Sometimes, having local storage and management may be more ideal, if it is only used locally, or there are only a few people using it, and only occasionally. Also, it depends on what type of information security is needed. If the material is highly sensitive, it may be better to use local storage only, and not allow outside access.

When planning for future growth, elasticity, or how quickly additional server capacity can be brought online and pay-for-service must be considered. Will the application need elasticity or is it something that is not accessed often or by many people? And if you are paying for services as you go, if there is a sudden surge of activity, will you be able to properly budget for that? If elasticity is not needed, or you want your costs to be fixed, then this must be considered when you are expanding your services.