

## **Experiences and Strengths:**

One of the most important skills I have developed throughout this course is containerization. I have also learned more about cloud development. Both skills will make me a more marketable candidate in my career field because it will allow me to embrace current technologies. Learning more about working with GitHub will also help me because it allows me to better collaborate with other developers.

My biggest strength as a software developer is my problem-solving abilities. Full-stack development involves figuring out how to approach problems at all points in development. Another strength I have as a software developer is my knowledge of the principles of object-oriented programming. Object-oriented programming languages like C++ and Python are among the most common languages used in many projects, and understanding the principles of object-oriented programming allows me to better adapt no matter what language I am coding in.

I feel I am prepared to assume a variety of roles in a new job. I feel that my work on this course has prepared me to work in full-stack development roles. This means I could also be prepared to work on front-end and back-end development as well as both.

## **Planning for Growth:**

Microservices and serverless approaches may help with producing efficiencies and scale in my work with web applications in the future. Container-based microservices are helpful for error handling because they allow for more isolation of code, which is helpful for troubleshooting. Serverless approaches are also helpful for scaling programs. For example, serverless approaches like those used in AWS often allow for automatic scaling of resources

according to need. This makes it easier to manage resources compared to other development approaches.

Predicting the cost of cloud services can be achieved in multiple ways. One example of this is in the scope of the project, which can be an effective indicator of how much development could end up costing. It is also important to consider labor costs related to software development, as well as other factors such as time and effort. It is also important to be able to adjust according to feedback from stakeholders.

I think that containers can be more cost-predictable than serverless programs. With serverless, you are often using a pay-for-use model. This means that when more resources are being used, the cost is higher. It also means that the cost can be lower when less resources are being used. With containers, the cost is more consistent. While there can be more potential for savings with serverless, I think the cost of containers can be easier to predict.

One of the biggest pros to expansion in cloud development is the potential to increase the reach of the product. This allows for more revenue potential. Depending on the business, it could also lead to better cost savings on certain resources, and it can also allow for more collaboration. However, one of the biggest cons is that expansion requires more resources, which requires more money. It can also lead to more limited control over business due to them not owning the infrastructure that they are using. There is also potential for more downtime if connectivity issues occur compared to more typical software development. Cloud development relies on the service provider always being up, and it can lead to negative consequences if there are moments when their service is not up.

Elasticity and pay-for-service both play a huge role in decision making for planned future growth. Elasticity in cloud development involves the ability to adjust resources according to many factors. This is important because it allows for scaling up and down according to changing needs. Pay-for-service is important because it allows for more efficient use of funds. It relates to elasticity because it is important to be able to either increase or decrease costs depending upon changing needs. Both elasticity and pay-for-service are useful because they allow for more efficient use of resources and money.