

Jonathan Vest

10/15/2023

CS470

Final Reflection

<https://youtu.be/xCWQcHacR0w>

- What skills have you learned, developed, or mastered in this course to help you become a more marketable candidate in your career field?

During this course we have learned about containerization and the tools used for containerization such as Docker. We learned about docker images and how to pull pre-built images from the docker registry. We learned how to create containers and how to create Dockerfiles. We used Docker Compose to orchestrate multiple containers as a logical unit. We also learned the proper steps when moving a full stack application to a web-native application. We took a close look at AWS web services and learned about storage services like S3 and EC2. We also learned how to utilize services such as AWS lambda and DynamoDB and how to secure applications and services utilizing IAM Roles and Policies.

- Describe your strengths as a software developer.

My key strengths as a software developer are that I understand software architecture and I have a solid technical background. I am good at prioritizing tasks, and I have the ability to handle multiple tasks at once. I have excellent communication skills, so I am able to break down complex problems and articulate the needs of a project to both colleagues and clients alike. I am a good team player and a quick learner. I also stay focused on the

needs of the end user, keeping their needs in mind as well as the needs of other stakeholders and the business.

- Identify the types of roles you are prepared to assume in a new job.

I feel prepared for roles in a software development capacity that will allow me to contribute as a member of a team while granting me the opportunity to learn from those with more knowledge and experience.

- Identify various ways that microservices or serverless may be used to produce efficiencies of management and scale in your web application in the future. Consider the following:

- How would you handle scale and error handling?

The principles of elasticity and Auto Scaling are integrated into AWS services. Elasticity is the ability to acquire as well as release resources when necessary. Amazon has several services that inherently offer elasticity such as Amazon S3. Services like EC2 and DynamoDB integrate with AWS Auto Scaling. AWS Auto Scaling monitors your applications and will automatically adjust capacity to maintain steady performance at the lowest cost. It can be difficult to trace errors across an environment and isolate their root cause. It is vital that teams understand how these services work together in order to optimize and troubleshoot serverless microservices.

- How would you predict the cost?

Most cloud services follow a pay-as-you-go model, in which you pay for what you have used, the run time of your service and the amount of data transferred. AWS offers a cost calculator that can be used to help determine the cost of specific services based on an estimated runtime and data transferred.

- What is more cost predictable, containers or serverless?

Services like AWS Lambda operate on a pay-as-you-go model. Microservices on containers require payment for the instance to run 24/7. While containers may not necessarily be the cheapest option, they would be more cost predictable as the cost would not fluctuate.

- Explain several pros and cons that would be deciding factors in plans for expansion.

Pros would include the ability to quickly deploy new code, services and resources. The pay as you go cost model would also be a pro when considering whether to expand as with most services you pay only for what you need or use.

Cons would include the somewhat limited control that you have with a serverless architecture. While serverless does abstract much of the painstaking maintenance away, this also means that we must release some control of those resources to the organizations that will be managing those tasks for us.

- What roles do elasticity and pay-for-service play in decision making for planned future growth?

Elasticity and the pay for service cost model offered by many cloud native computing organization such as AWS will play a key role in the decision to expand applications.

Elasticity is the ability of a service to acquire or release resources as necessary. This means that our applications are free to grow without having to worry about underperformance if our service sees a large increase in users or unused overhead if we were to acquire resources and have them sit idle. This is also true with the pay as you go or pay for service cost model. This model charges only for the runtime of our service or

for the amount of storage that we need. We again don't need to worry about unused overhead or being charged for resources or storage space that are sitting idle.