Michael Cannon Southern New Hampshire University

Full Stack Development II Friday, December 16, 2022

Reflection

https://youtu.be/rpC5XS5YtNk

This course has helped me to understand modern cloud application development and how the elasticity of storage and rapid deployment of containerized microservices is the best way to create scalable, always on cloud applications that can also be served globally. My strength as a developer has always been to absorb novel information in the form of APIs, requirements, syntactical languages, and concepts. One concept that I have not encountered often is precisely what this class covered: modern cloud computing for web services. With this new knowledge I am prepared to create and develop containerized applications that can be deployed as needed with the flexibility of being system-agnostic, making them ideal for on-premises or cloud implementation.

Microservices and serverless architectures can be used to produce efficiencies of management and scale in web applications in several ways. Microservices and serverless architectures allow for greater scalability and improved error handling because they decouple the different components of the application. This means that if one component experiences an error, it can be isolated and fixed without affecting the rest of the application. Additionally, microservices and serverless architectures can automatically scale up or down based on demand, allowing the application to handle fluctuations in traffic without the need for manual intervention.

It can be difficult, however, to predict the cost of using microservices or serverless architectures as they are based on usage and can vary depending on the specific workloads and resources required. The most crucial component to effective cost estimation is having a realistic figure of both the storage requirements and the anticipated utilization, which can be used to roughly estimate operational costs. Microservices and serverless architectures can offer cost savings compared to traditional architectures because they only charge for the resources used, rather than requiring the use of a fixed infrastructure.

Containers and serverless architectures can be cost-effective options for managing and scaling web applications. Containers are typically more cost-effective for applications with steady, predictable workloads because they allow for the allocation of fixed resources. Serverless architectures, on the other hand, are typically more cost-effective for applications with variable or unpredictable workloads because they only charge for the resources used.

There are several key benefits to consider when deciding on plans for expansion using microservices or serverless architectures. Microservices and serverless architectures allow for easy scalability of individual components, which can be useful for handling sudden spikes in traffic or demand. They also can provide improved fault tolerance because due to decoupling the different components of the application. This means that if one component experiences an error, it can be isolated and fixed without affecting the rest of the application. Both architectures allow for greater flexibility in terms of the technology stack used and the ability to easily update and deploy new features.

Despite these benefits, Microservices and serverless architectures have some drawbacks, and most of them are directly or indirectly related to cost. Microservices and serverless architectures can be more complex to implement and manage compared to monolithic architectures. This can require additional resources and expertise to set up and maintain, more expertise means more potential overhead costs for employment. Integrating different microservices or serverless functions can be challenging because they may require different technologies and development approaches. Finally, these approaches can be more expensive to implement and maintain compared to monolithic architectures, especially if the application requires a large number of services or functions. Overall, the pros and cons of using microservices or serverless architectures for expansion will depend on the specific needs and goals of the application and the resources available for implementation and maintenance.

Elasticity and pay-for-service are two crucial factors to consider when making decisions about planned future growth using microservices or serverless architectures.

Elasticity refers to the ability of an application to scale up or down in response to changes in demand. Both microservices and serverless architectures can offer elasticity because they allow for the scaling of individual components or functions based on usage. This can be useful for handling sudden spikes in traffic or demand and can help to reduce costs by avoiding the need to provision resources that are not being used.

Pay-for-service refers to the billing model used by microservices or serverless architectures, which typically charges based on the resources used. This can be more cost-effective compared to traditional monolithic architectures, which often require the use of a fixed infrastructure regardless of usage. Pay-for-service can be a deciding factor in planned future growth because it allows for more flexible and predictable costs based on the specific needs of the application. Overall, elasticity and pay-for-service can be useful considerations when making decisions about planned future growth using microservices or serverless architectures because they allow for greater flexibility and cost-effectiveness in terms of scaling and resource usage.