Serverless, a cloud-native approach to building APIs

**C768, Task 1**

**Robert J. Robinson**

**01168728**

**Western Governors University**

Abstract

In the current world, technology has improved with modern cloud-first solutions. Most companies have embraced the technology to get to market faster, scale without complications, and offload expensive critical infrastructure. The paper has demonstrated the products provided by Amazon Web Services, such as AWS Lambda. It has well explained the serverless computing architecture and the serverless application model with its benefits in empowering modern organizations with cloud first approaches. A discussion in creating a new API is well articulated with less dependency on infrastructure to attain on-demand scalability that serves our organization. When building our public API, the focus is the ability to make it quickly and scale it on demand. Using a combination of API Gateway and AWS Lambda, it creates small parts of the API while focusing on quality. The paper shows how the API Gateway allows a Lambda to process and respond to the HTTP request. It explains the Serverless Application Model, SAM, and the accompanying command line interface tool that provides a simple syntax for event triggers and provisioned resources.

Introduction

Cloud-first solutions have grown to be what the modern companies view as a primary resource to get to market faster, scale with less effort, and offload expensive procurement of critical infrastructure. The most established company in the provision of these cloud services is Amazon Web Services. One of the products they offer is AWS Lambda that permits infinitely scalable compute processes and gives any company the ability to offload processing to these services at a fraction of the cost with minimal training. Lambda is described an event-driven serverless computing platform that links to several other services by AWS. Supported by the Serverless Application Model (SAM), an open source-framework that allows a simple syntax and command line tool to express functions, APIs, and event source mappings. Lambda, when used in conjunction with other AWS services, provides the compute and logic layer of an application. This layer, a micro-service event-driven application, is dispersed in the entire AWS footprint. An example application of the technology is the ability to build a REST API and not bothered with a particular programming language to use, a lengthy deployment time, as well as limiting over or under capacity planning (Rajan, 2020).

The AWS Lambda is a serverless computing platform, and some times referred as a Function as a Service (FaaS). Developers who embrace this platform are focused more on the product and business logic rather than responsibilities such as the operating system, access control, operating system patching, scaling, right-sizing, provisioning, and availability in a traditional approach. Going serverless allows developers to write and execute code without configuration management of the underlying servers. AWS handles the entire compute lifecycle as well as infrastructure in the background, and the developer handles the logic located in the function. While you may lose some flexibility in various respects, applications/functions are only billed when the code is executed. If the user uses 1 ½ seconds of computer time, that is all of what they had billed. Lambda's best activity or practice is one function that only handles one type of request or an event trigger (Crane & Lin,2017). The lambda function then accepts a JSON payload and will either respond to the requestor or pass along the payload to another service, and then the function completes. Most operations run in milliseconds, but there is the ability to extend process time to a maximum of five minutes. Another fantastic feature of this service is that this function is spread across many locations allowing for concurrent requests to be handled without worrying about a traditional server overload. Other AWS services are the ones that start an event trigger, such as the API Gateway, the AWS routing service. API Gateway allows a lambda to process and respond to the HTTP request.

As we start to develop our public API, careful consideration should be made on what technology stack to use. Following the Serverless Application Model, AWS has outlined a set of best practices that allows for a quick and efficient deployment across many deployment stages, and an iterative approach. Using the SAM CLI, a simple syntax allows for the developer to wire up a completely distributed application across various services, with minimal configurations.   
This CLI can cut deployment and provisioning time to a fraction of a traditional approach and save considerably on capital expenses. This CLI tool not only creates, but will also tear down provisioned resources in the event that hey are no longer used in the application.

Conclusion

Ways of building a new API have been discussed with less dependency on infrastructure and attaining on-demand scalability that saves the company money. Using an AWS Lambda model is advantageous and beneficial to any organization. The creation and utility of the serverless applications on AWS relieve developers some responsibilities and constraints introduced by managing servers. It also relieves the company of some of the expenditure burdens through minimizing the workforce, perhaps a dedicated system administrator, that maximizes profit. The utility of Lambda as the serverless logic layer allows the organization to expand its growth and focus on its development efforts that differentiate the company from competitors. Alongside Lambda, the AWS offers additional serverless capabilities that can assist in building robust, event-driven, performant, secure, reliable, and cost-effective applications. Understanding the capabilities and suggestions described in the white papers can help ensure the company's success when building serverless applications for an organization. As an example, a standard Rest API traditionally a massive monolith, is slow to change, difficult to scale, and expensive. Selecting this technology stack allows the focus of the organization to be able to build quickly without worrying about how it could impact stakeholders or customers. Using API Gateway together with Lambda, small sections of API are created at a time with a strong focus on quality. The SAM CLI tools brings all of the services together and provides the team a simple interface, automates the provisioning and deprovisioning of resources, and speeds up mean time to deployment. These actions have a direct value of saving the company financial recourses that could be put else where.

References

Crane, M., & Lin, J. (2017, October).An exploration of serverless architectures for information retrieval.In *Proceedings of the ACM SIGIR International Conference on Theory of Information Retrieval* (pp. 241-244). Retrieved on 22nd October 2021 from: https://dl.acm.org/doi/abs/10.1145/3121050.3121086

Rajan, R. A. P. (2020). A review on serverless architectures-function as a service (FaaS) in cloud computing. *Telkomnika*, *18*(1), 530-537. Retrieved on 22nd October 2021 from:https://www.proquest.com/openview/d54a70a969941a357ad3aef698bb030b/1?pq-origsite=gscholar&cbl=376296