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 their market faster, scale without complications, and offload expensive critical infrastructure procurement. The paper has demonstrated the products provided by Amazon Web Services, such as AWS Lambda. It has well explained the serverless computing architecture with its benefits in running organizations in the current world. A discussion of means of building a new API is well articulated with less dependency on infrastructure to attain on-demand scalability that serves present organizations. In building a public API, the focus for the company is the ability to make it quickly. It has used a combination of API Gateway and Lambda that 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 demonstrates the way AWS offers a command-line tool known as Serverless Application Model (SAM). The SAM tool allows the developer to build a cloud formation that is useful in deploying any AWS service as well as the corresponding event triggers.

Introduction

In the present days, cloud-first solutions have grown to be what the tech companies take as the primary resource to get a market faster, scale with less effort, and offload expensive procurement critical infrastructure. The most established company in the provision of these services is Amazon Web Services. One of the products they offer is AWS Lambda that permits infinitely scalable computer processes and should give any company the ability to offload processing to these services at a portion of the cost. Lambda is known to be an event-driven serverless computing platform that links to several other services by AWS. It also defines Lambda as the serverless application model (SAM). It is a micro-service event-driven application that is dispersed in the entire AWS footprint. A single application of the technology is the ability to build a public 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 known to be a serverless computing platform. The developers of the forum are much focused on the main product as well as business logic rather than responsibilities such as the operating system, access control, operating system patching, scaling, right-sizing, provisioning, and availability. The real meaning of serverless is that the developers can write and execute code without configurations or management of the underlying servers. The main focus of the developer is to write the coding. The AWS handles the entire compute as well as infrastructure in the background. While you may lose some flexibility in various respects, the applications are 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 part 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. The other AWS services are the ones that start the triggers, such as the API Gateway. It is a routing service that allows a lambda to process and respond to the HTTP request. AWS also provides a command-line tool called SAM. The same tool permits the developer to create a cloud formation template with provisions for deploying any AWS service as well as the corresponding event triggers(Crane & Lin,2017).

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. The standard Rest API, a massive monolith, is slow to change, difficult to scale, and has inadequate testing. The public API was rebuilt, and the focus for the organization should be the ability to build quickly without worrying about how it could impact stakeholders. Using API Gateway together with Lambda, small sections of API are created at a time based on quality.

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