



Serverless Web App in AWS Cloud

Pankaj Kumar Singh¹, Md Sahajad Ansari², Akula Leena Keerthi³, Mrs Akhila Nalla⁴

^{1,2,3,4}Department of CSE, Aditya Engineering College, Surampalem, A.P., India

ABSTRACT:

Building serverless operation means you can concentrate on core operations. Serverless technologies features automatic scaling, erected- in high vacuity and a pay-for- use billing model to increase dexterity and optimize cost. These technologies also count structure operation tasks like capacity provisioning and doctoring. Serverless operation starts with AWS Lambda, an event- driven cipher services natively integrated with over 200 AWS services and SaaS. AWS manages all the factory without managing garçon.

INDEX TERMS: Serverless Computing, Amazon Web Services

I. Introduction:

In this design we have created a simple serverless website that helps user to request the feedback form for council check. Serverless is a process which shows services, practices and strategies which is used to make a website so as to introduce and develop for parlous changes. Serverless computing contains structure operation tasks as capacity provisioning and doctoring. formulators can easily concentrate on developing law that serves the customer. serverless services like AWS lambda has a quality of machine scaling and pay as per use billing model. Serverless structure does not bear any waitpersons. formulators can concentrate on core product and a business sense. Serverless operations or website doesn't bear you to manage any different waitpersons. We can easily concentrate only on core product and business sense, rather of fastening on the arrears like operating system(zilch's) access control, zilch's doctoring, provisioning, right- sizing, scaling, and vacuity. Serverless computing offers a number of advantages over traditional pall- predicated or garçon predicated structure. Serverless architectures offer high scalability, high strictness, and quicker time to release, all at a reduced cost. thus, formulators do not need to worry about purchasing, provisioning, and managing backend waitpersons. still, serverless computing is not a magic bullet for all operation inventors. To make a web operation using the following AWS Services AWS DynamoDB as the database AWS Lambda to create functions that will read and write from/to the database.

AWS Lambda to produce functions that will read and write from to the database.

AWS API Gateway to produce the REST API that the web operation will use.

AWS S3 to host the web operation.

AWS CloudFront to deliver the web operation from a position near to the stoner's position.

Amazon Cognito to give Authentication.

Amazon SNS to give messaging service for communication and announcement.

Amazon IAM to specify who can pierce the services.

Frontend (HTML, CSS, JavaScript) to produce and design the web runners

II. Objective

Serverless computing offers a number of advantages over traditional pall.

- Predicated or garçon predicated structure.

-Serverless architectures offer high scalability, high strictness, and quicker time to release, all at a reduced cost. thus, formulators do not need to worry about purchasing, provisioning, and managing backend waitpersons.

- Still, serverless computing is not a magic bullet for all operation formulators.

III. On AWS Platform

You can make a serverless website by using different AWS services. Each service is fully managed and it is not bear to manage waitpersons.

You have to configure them and upload the law to AWS Lambda, which is also called as serverless cipher service.

1. S3 Storage Service (S3 bucket)

Amazon S3 stands for Simple Storage Service. Amazon S3 stores the data in buckets which have capacity of maximum 100 buckets and is used to store and recoup any amount of data from anywhere analogous as web spots and mobile apps, marketable operations and data from IoT sensors and bias. We produce and upload a html train of our feedback form. From that an Object URL is generated from that we can pierce the website and submit the review.

2. API Gateway

An API gateway is an API operation tool to connect a customer and a collection of backend services. An API gateway used to pierce the HTTP and HTTPS to the lambda function.

3. AWS Lambda

In this, API gateway triggers the lambda function and the reviews will store into the table named reviews.

4. AWS DynamoDB

The reviews submitted by the scholars will be stored in the table which we've created in the AWS DynamoDB named as reviews.

IV. Advantages

Garçon operation is not necessary. There is not necessary to manage any garçon as we have created a serverless website. Pay as per the use, reduce cost. In AWS it has a quality of pay as per the use that means we just have to pay only for that an AWS service which we use to produce a serverless website.

1. Scalable

Scalability is one of the quality of AWS cloud in which we can fluently gauge up or gauge down the space as per our conditions.

2. Quick updates are possible

By using serverless structure or an armature it's easy to emplace or modernize the web operation. This property is generally used by the app inventors.

V. Disadvantages

Testing and debugging come more gruelling . In AWS after using the serverless armature testers face further difficulties to test and remedy their web operation on AWS cloud platform.

1. New security concern

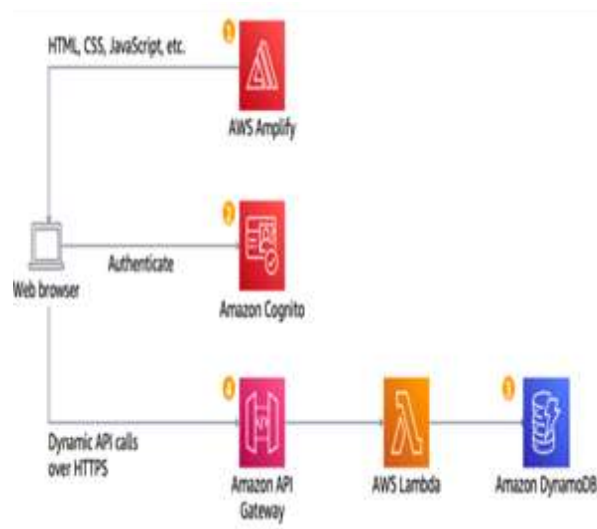
For the inventors working on a backend face more problems regarding security of a law or any other important effects which is important to secure.

2. Not for long-running processes

In AWS pall platform the serverless armature isn't made for long term process or we can call as long handling processes.

VI. Conclusion

Structure serverless operations on AWS shows that the arrears that waitpersons introduce. Using AWS Lambda as our serverless sense caste used to make hastily and concentrate our development sweats on what differentiates our website. Lambda, AWS provides fresh serverless capabilities so that we can make robust, reliable, secure, and cost effective website. Understanding the capabilities and recommendations described in this disquisition paper can help to ensure our success when erecting serverless website of our own.



VII. REFERENCES

1. <https://aws.amazon.com/getting-started/hands-on/build-serverless-web-app-lambda-apigateway-s3-dynamodb-cognito/>
2. <https://ieeexplore.ieee.org/document/9058237>
3. <https://ijert.org/papers/IJERT2107116.pdf>
4. https://ijirt.org/master/publishedpaper/IJIRT151299_PAPER.pdf

VII. Project Screen Shots

