Project Report

on

“Serverless Contact Form using AWS”

Project Overview:

Project Name: serverless Contact Form using AWS

Created By : Rushikesh Rajesh Waghule

Project Duration: 10-11-23 to 18-11-23

Executive Summary:

The Serverless Contact Form project aims to create a scalable and cost-effective solution for handling contact form submissions on a website using Amazon Web Services (AWS) serverless services. By leveraging AWS Lambda, API Gateway, and Simple Email Service (SES), this project ensures a reliable and scalable architecture without the need for traditional server infrastructure.

Objective: To build a serverless contact form that allows website visitors to submit inquiries or messages.

Technologies Used:

AWS Lambda

API Gateway

Simple Email Service (SES)

AWS Identity and Access Management (IAM)

AWS CloudFormation

Architecture:

The project utilizes the following AWS services:

Lambda Function:

Receives HTTP POST requests from the contact form on the website.

Processes the form data and sends it to SES for email delivery.

API Gateway:

Acts as a trigger for the Lambda function, providing an HTTP endpoint for the contact form.

SES (Simple Email Service):

Sends email notifications containing the form data to the specified email address.

IAM (Identity and Access Management):

Defines the necessary permissions for Lambda to invoke SES and logs to Amazon CloudWatch.

CloudFormation:

Automates the deployment and management of AWS resources.

Deployment:

Deployment is automated using AWS CloudFormation. The template includes the Lambda function, API Gateway, IAM roles, and necessary permissions. This allows for easy replication of the project in different AWS environments.

Conclusion:

The Serverless Contact Form project provides a scalable, cost-efficient, and easily deployable solution for handling contact form submissions. By leveraging AWS serverless services, the project ensures high availability and eliminates the need for maintaining traditional server infrastructure.

Future Enhancements:

Add validation to the Lambda function to ensure data integrity.

Implement AWS CloudWatch Alarms for monitoring and alerting.

Explore additional features such as file uploads or CAPTCHA for security.

Project Timeline:

The project will be executed in the following phases:

Planning and Setup:

Define project scope and objectives.

Set up AWS accounts and access permissions.

Plan S3 bucket configuration and custom domain choice.

Implementation:

Create the S3 bucket and configure for website hosting.

Develop the simple HTML/CSS webpage.

Upload files to the S3 bucket.

Configure the S3 bucket policy.

Conclusion:

This project aims to demonstrate the process of hosting a static website on AWS S3. By following the outlined steps, i expects to achieve a seamlessly hosted website with a custom domain accessible to users.

Acknowledgments:

The project team acknowledges the support and resources provided by AWS documentation, forums, and community. =