Building a Cloud-Native portfolio

AWS HTML Website with Serverless Architecture

By Alberto Gómez, AWS Certified Solutions Architect, December 2023

Table of contents

1. Introduction.

What better way for a Cloud Solutions Architect to demonstrate its skills than by creating a resume on the cloud rather than sending casual PDF's just naming competencies. This way of thinking coupled with my firm conviction in the efficacy of hands-on learning aroused the creation of my own CV using the AWS ecosystem.

My journey began by taking the practical step of hosting my personal website in S3 (Simple Storage Service) as a learning exercise. However, the idea evolved with the study of services and techniques acquired during my study to become an AWS certified Solutions Architect. So, as time passed, I kept adding new features to learn at firsthand what I was studying.

This project not only showcases technical capabilities, but also adheres to the principles of cloud-native architecture. The portfolio seamlessly integrates various AWS services to enhance the interactivity and functionality of the website, using serverless technology exclusively, to achieve scalability, efficiency, and cost-effectiveness.

2. Project Overview.

The aim of this project is to create a cloud-native static website hosted on AWS S3, leveraging serverless services for various functionalities. The project displays a comprehensive understanding of AWS services, focusing on the serverless paradigm to ensure optimal resource utilization and ease of management.

One of the distinguishing features of this project lies in the implementation of an advanced "Contact Me" form, which uses API Gateway to trigger a Lambda function. This Lambda function extracts information submitted through the form's JavaScript code and forwards it securely to my third-party email account via AWS Simple Email Service.

3. Key Components.

Let us begin with the different services I decided to use for this project:

- **AWS S3:** a highly durable and scalable object storage service that serves as the foundation for hosting the static website. All website assets, including HTML, CSS, JavaScript, and multimedia files, are stored in a S3 bucket.
- **AWS CloudFront** for Content Delivery: Amazon CloudFront, a content delivery network (CDN) service, is used for efficient and low-latency content distribution. By caching content at edge locations worldwide, CloudFront enhances website performance and reduces latency for end-users around the world.

- **AWS Route 53** employed as the Domain Name System (DNS) provider to route incoming traffic to the website. This service enables seamless management of domain names, and it integrates flawlessly with other AWS services.
- **AWS Certificate Manager** for SSL/TLS: to ensure secure communication between the user and the website. This enhances the website's security by encrypting data in transit.
- **AWS API Gateway**: used for creating a RESTful API endpoint that acts as a bridge between the static website and the Lambda function.
- **AWS Lambda** for Serverless Logic: triggered by API Gateway when the "Contact Me" form is submitted. This function processes the form data, ensuring efficient handling of user inputs.
- **AWS SES** for Email forwarding to my third-party email account. SES ensures reliable and scalable email delivery with built-in security measures.

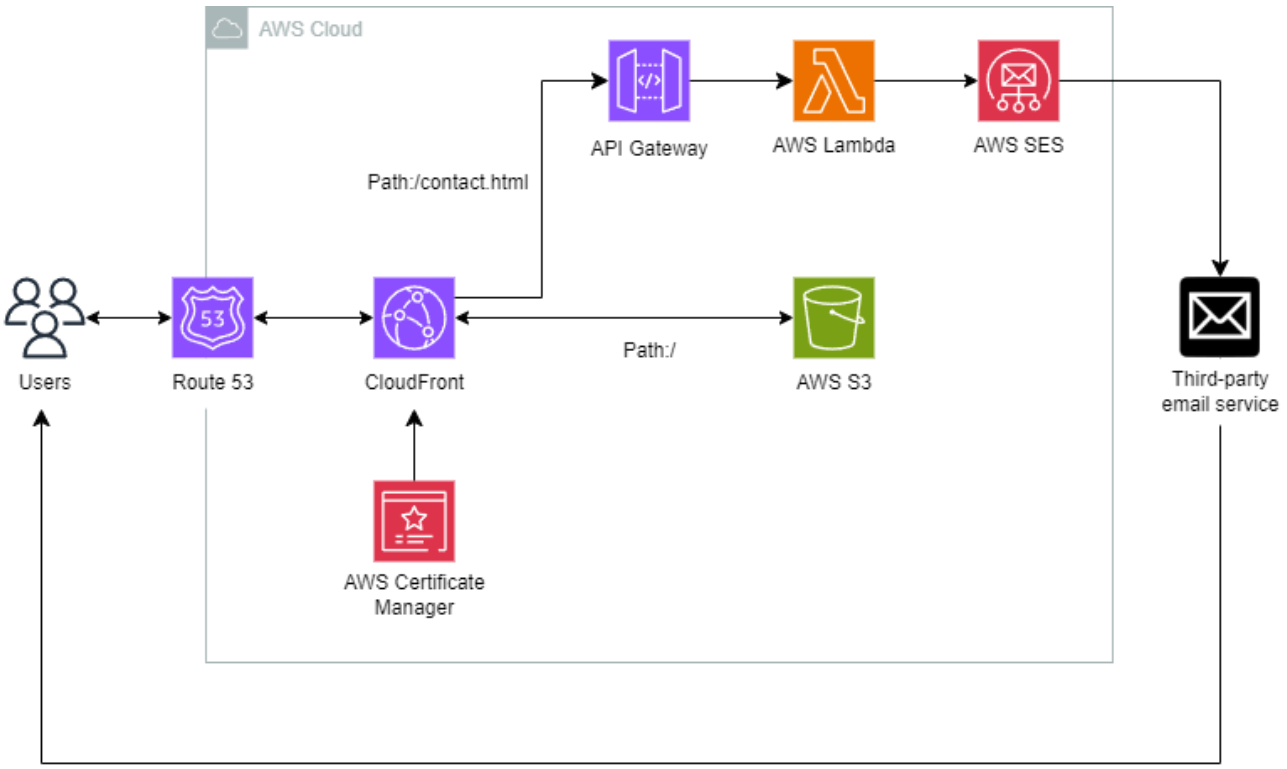
Using the static website hosting functionality within an S3 bucket allowed me to host my portfolio for less than two dollars per month, while also enabling seamless scalability to accommodate hundreds or thousands of users, while CloudFront reduced the media assets loading time by catching the content in the AWS Edge locations that are located around the world.

Additionally, the services integrating the contact form (API Gateway, Lambda and AWS SES) are billed based on the number of requests and the duration they take. So, I only pay when actually using the services. In total, less than 0.4 USD per month based on the emails I receive.

Altogether, an AWS cloud-native portfolio is a powerful representation of my ability to harness the full potential of cloud computing. The advantages of scalability, cost optimization, flexibility, and security, coupled with proven skills in the different AWS services, make this project a valuable asset in my CV.

4. Architecture design.

For an easier construction and integration of all components I designed the architecture's diagram with the services I wanted to use:



One a user has written my DNS in its browser is directed to Cloudfront who presents the SSL certificate and cyphers the connection. After that, the user goes straight to the index.html hosted on the S3 bucket.

If someone clicks the submit button in the contact form, the API Gateway automatically triggers a Lambda function that reads the JS code and extracts all information submitted by the user in the form and forwards to SES who sends the message.

5. Benefits of Cloud-Native and Serverless architecture.

Serverless architecture allows for automatic scalability, ensuring that the website can handle varying levels of traffic efficiently without the need for manual intervention. Additionally, serverless services minimize costs associated with idle resources. I only pay for the actual compute time and resources consumed during execution. Not to mention the ease of management since these services take away all infrastructure maintenance.

Furthermore, the integration of API Gateway and Lambda allow for automatic scaling based on demand, ensuring that the contact form can handle varying levels of submissions without

manual intervention while facilitating real-time communication between the static website and the serverless backend, ensuring a seamless user experience.

Finally, AWS Simple Email Service enhances the security and reliability of the contact form's email functionality. SES provides a trusted and scalable solution for forwarding user messages to my third-party email account.

6. Conclusion.

The development of an AWS cloud-native portfolio is a powerful representation of the ability to harness the full potential of cloud computing. The integration of AWS Lambda, CloudFront, Route 53, Certificate Manager, API Gateway, and AWS SES proves a holistic understanding of the different AWS services. This project not only highlights technical proficiency but also emphasizes the importance of building applications that are scalable, cost-effective, and easy to manage in a cloud-native environment. As cloud computing continues to shape the future of technology, the skills demonstrated in this project are invaluable for those who aspire to excel in the cloud domain.