Serverless Portfolio Website with Contact Form (AWS Free Tier Project)

Project Overview

This project showcases how to build a modern, fully serverless portfolio website using AWS Free Tier services. The goal is to create an interactive, cost-effective portfolio that allows visitors to connect via a contact form--without needing to manage any backend infrastructure.

The website is completely static and hosted on Amazon S3. Dynamic features, like the contact form, are powered by serverless tools such as API Gateway, Lambda, and SES. CloudFront is used to deliver the site quickly and securely across the globe. This project is primarily a demonstration, designed to highlight hands-on AWS skills and cloud proficiency.

Problem Statement

Typically, setting up a portfolio with a working contact form requires backend servers, custom code, and third-party mailing services. These elements can be complex, expensive, and time-consuming to maintain.

This project solves that problem using AWS serverless services, eliminating the need for server management and minimizing costs--especially for students or beginners working within the AWS Free Tier.

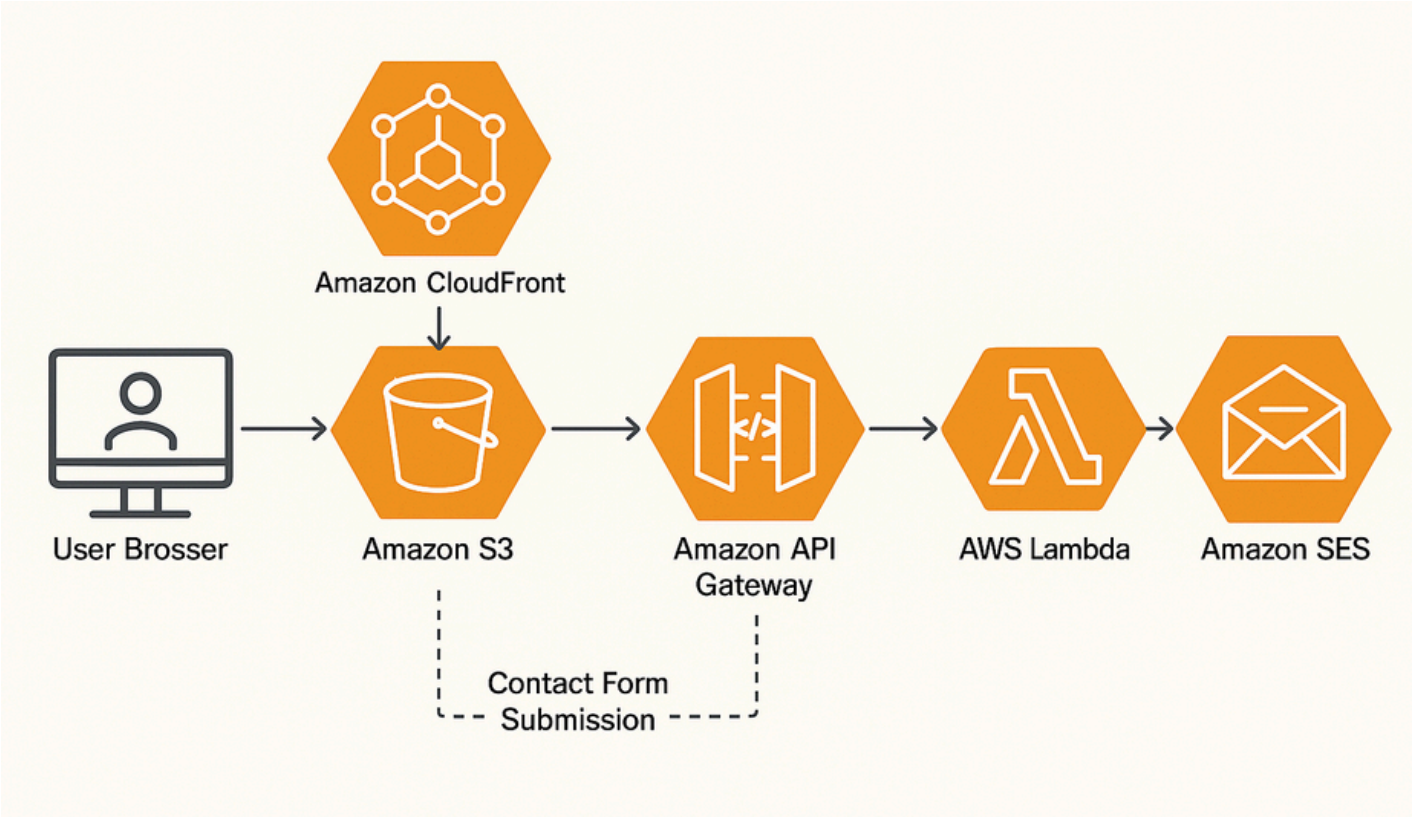
Solution Summary

This serverless architecture is built using the following AWS services:

- Amazon S3 -- Hosts the static HTML, CSS, and JavaScript files.
- Amazon CloudFront -- Distributes the site globally with HTTPS and fast content delivery.
- Amazon API Gateway -- Accepts and handles secure form submissions.
- AWS Lambda Processes the form data without a dedicated server.
- Amazon SES (Simple Email Service) Sends the message to the site owner via email.
- IAM (Identity and Access Management) Secures communication between Lambda and SES.

How It Works (Architecture Flow)

- A user visits the portfolio website through a custom domain or S3 link.
- CloudFront serves the static site files using HTTPS.
- The user fills out the contact form and clicks "Submit."
- The form data is sent as a POST request to API Gateway.
- API Gateway triggers a Lambda function.
- Lambda validates the input and prepares the message.
- Lambda then uses SES to send the message to the site owner's email.
- Finally, the user receives a success message on the website.



Tools & Services Used

- Amazon S3 -- For static website hosting
- Amazon CloudFront -- For global content delivery and HTTPS support
- Amazon API Gateway -- For securely handling form submissions
- AWS Lambda -- For running backend code without managing servers
- Amazon SES -- For sending email notifications
- IAM -- For secure role-based access and permissions

Benefits

- No server to manage -- Completely serverless architecture
- Free Tier-friendly -- Minimal or zero cost for most users
- Globally scalable -- CloudFront ensures fast delivery anywhere
- Secure and efficient -- HTTPS, IAM roles, and managed services
- Great for learning -- Practical experience with AWS tools

Conclusion

This project is a great starting point for anyone interested in cloud development. It combines static hosting with serverless backend features to deliver a professional portfolio--without ongoing infrastructure costs. Whether you're showcasing it on LinkedIn, GitHub, or during interviews, it's an excellent way to highlight your AWS knowledge.