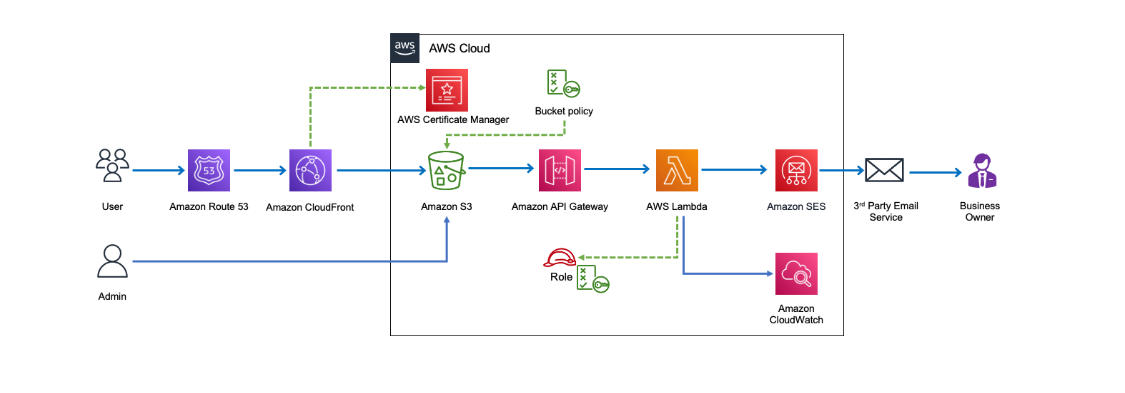
# Serverless Contact Form Architecture – Step-by-Step Implementation

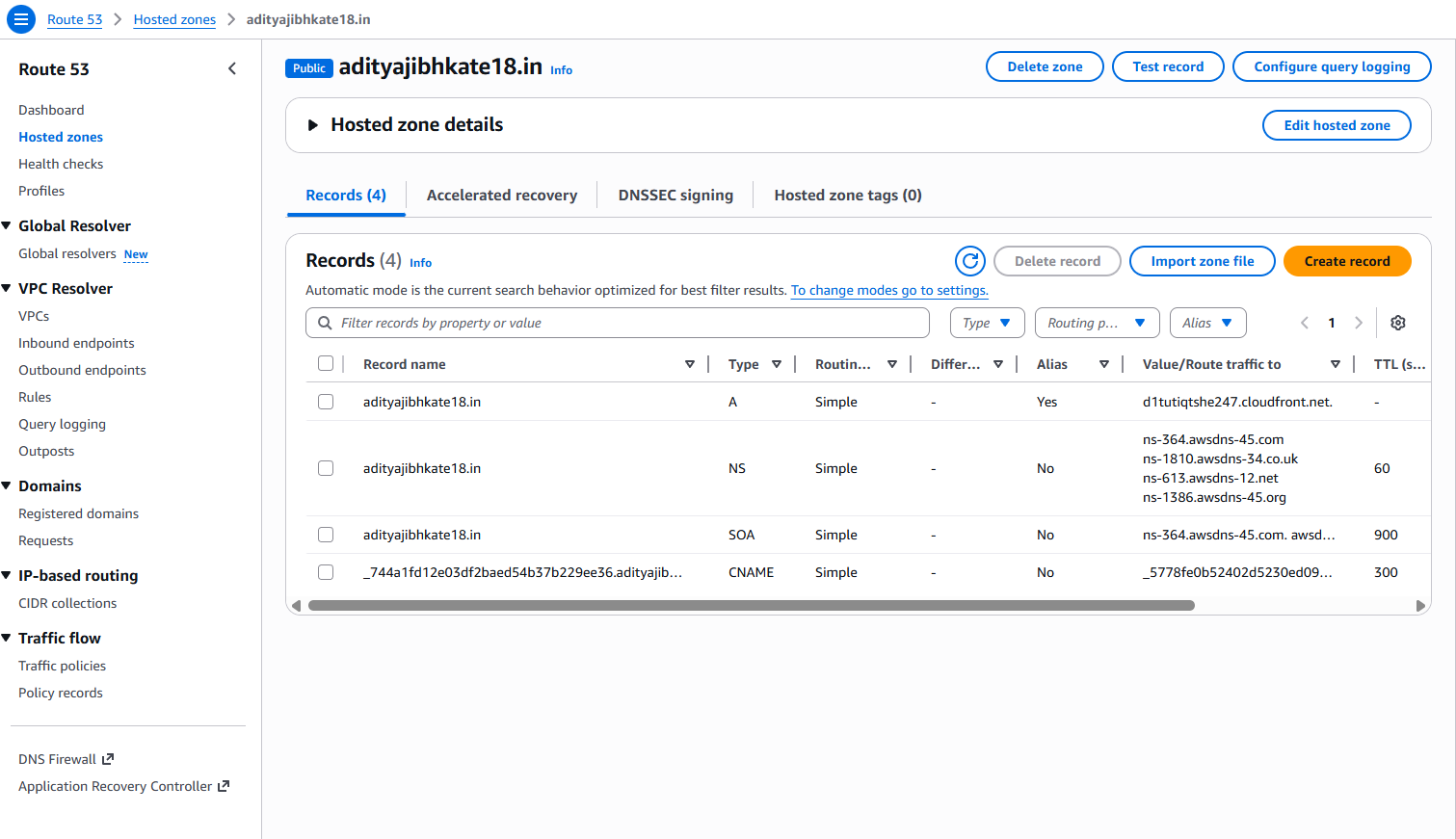
## 1. Project Overview

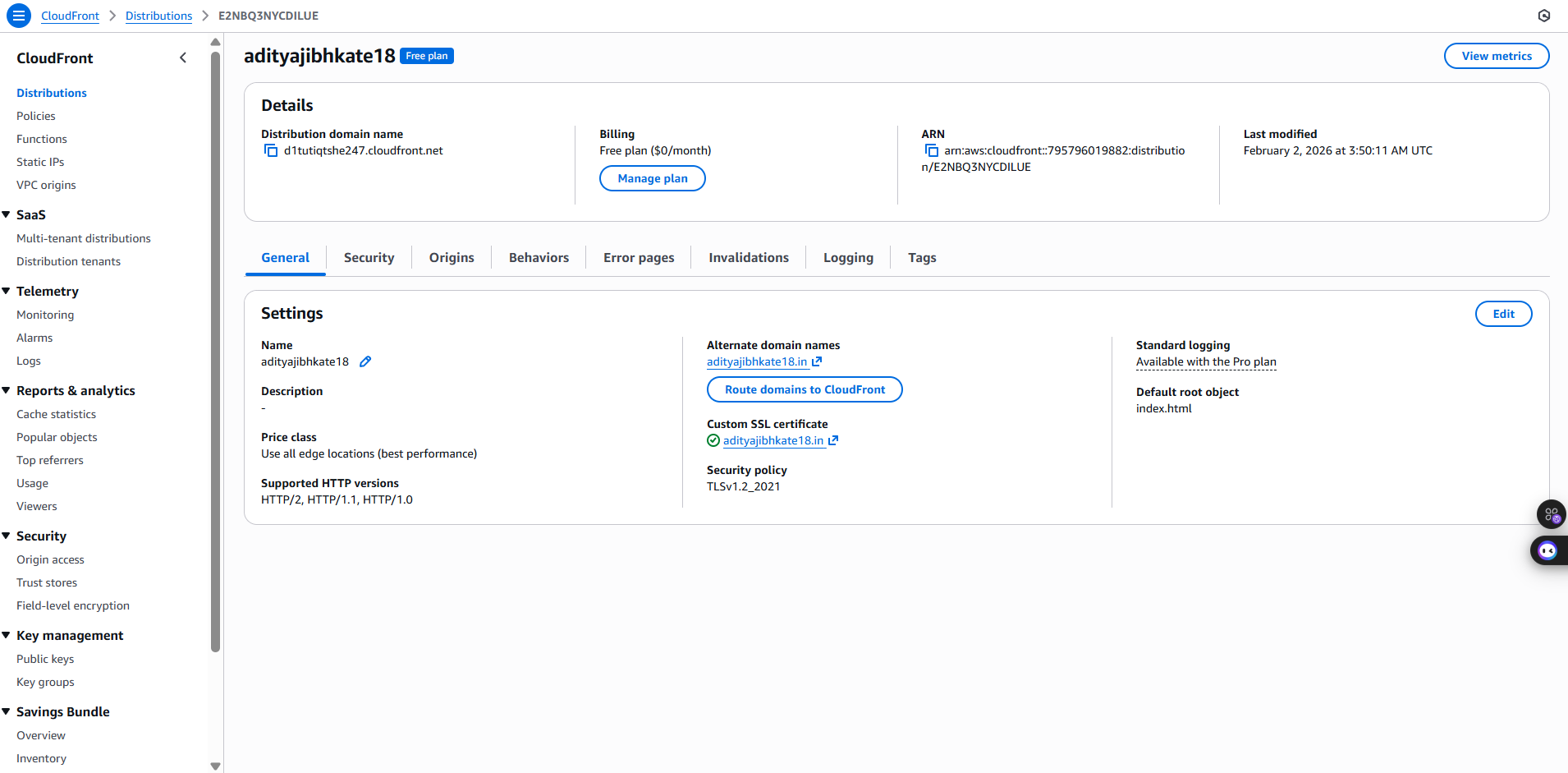
This document explains the step-by-step implementation of a fully serverless contact form using AWS services. The solution hosts a static website, processes form submissions, and sends email notifications without managing servers.



## 2. Domain & DNS Setup (Amazon Route 53)

• Purchased or used an existing domain in Route 53.  
• Created a hosted zone.

  
• Configured A record pointing to CloudFront distribution.

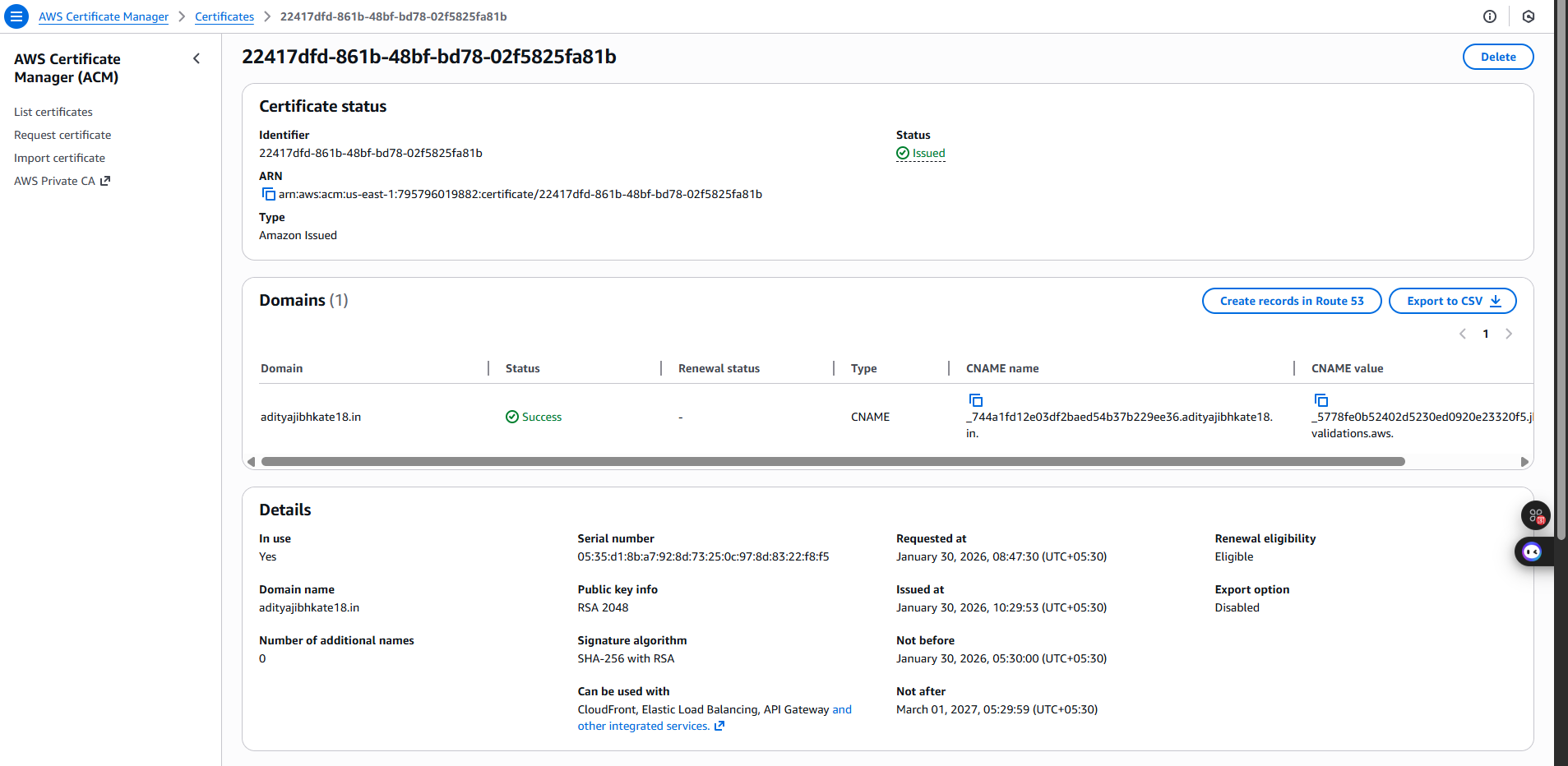


• Enabled DNS routing for high availability.



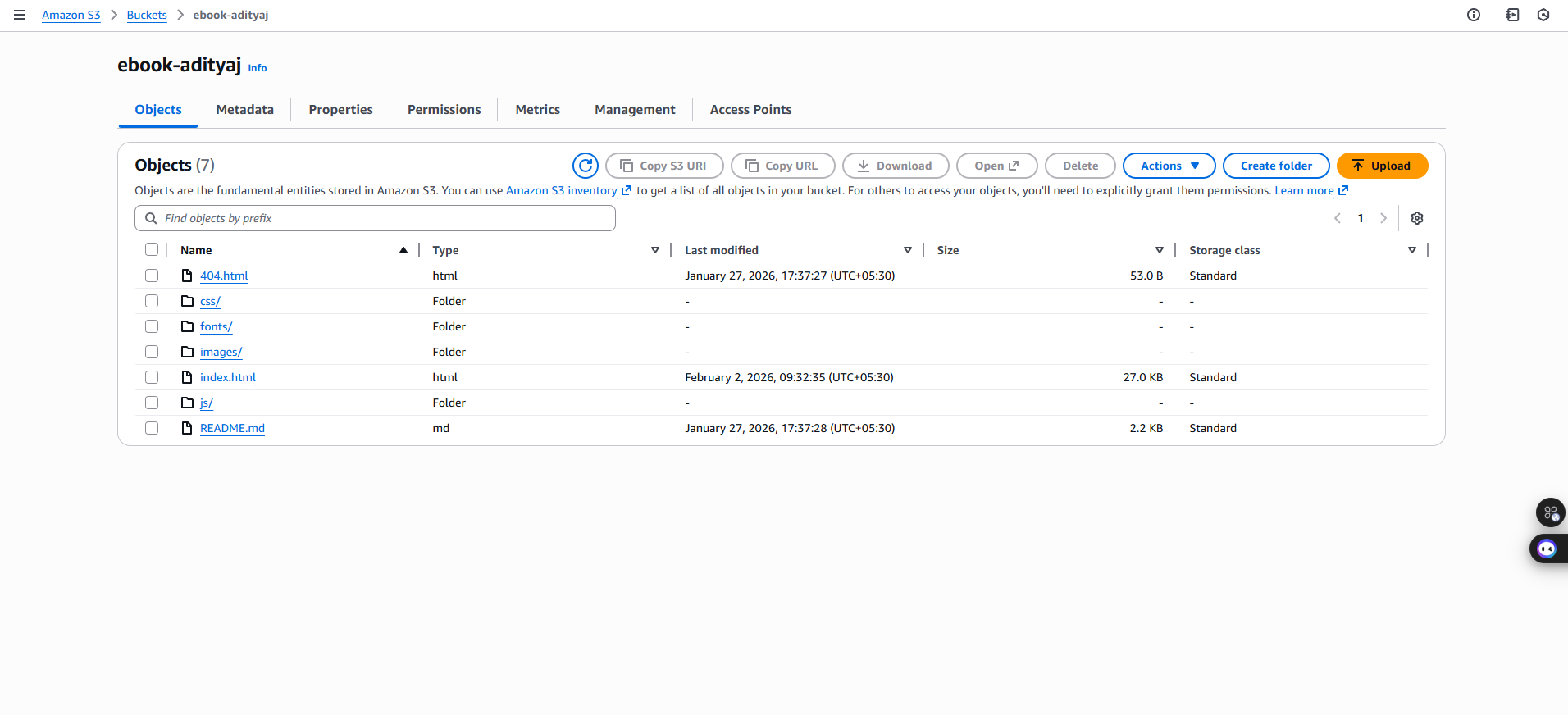
## 3. SSL Certificate Setup (AWS Certificate Manager)

• Requested a public SSL certificate.  
• Validated domain ownership using DNS validation.  
• Attached certificate to CloudFront for HTTPS access.



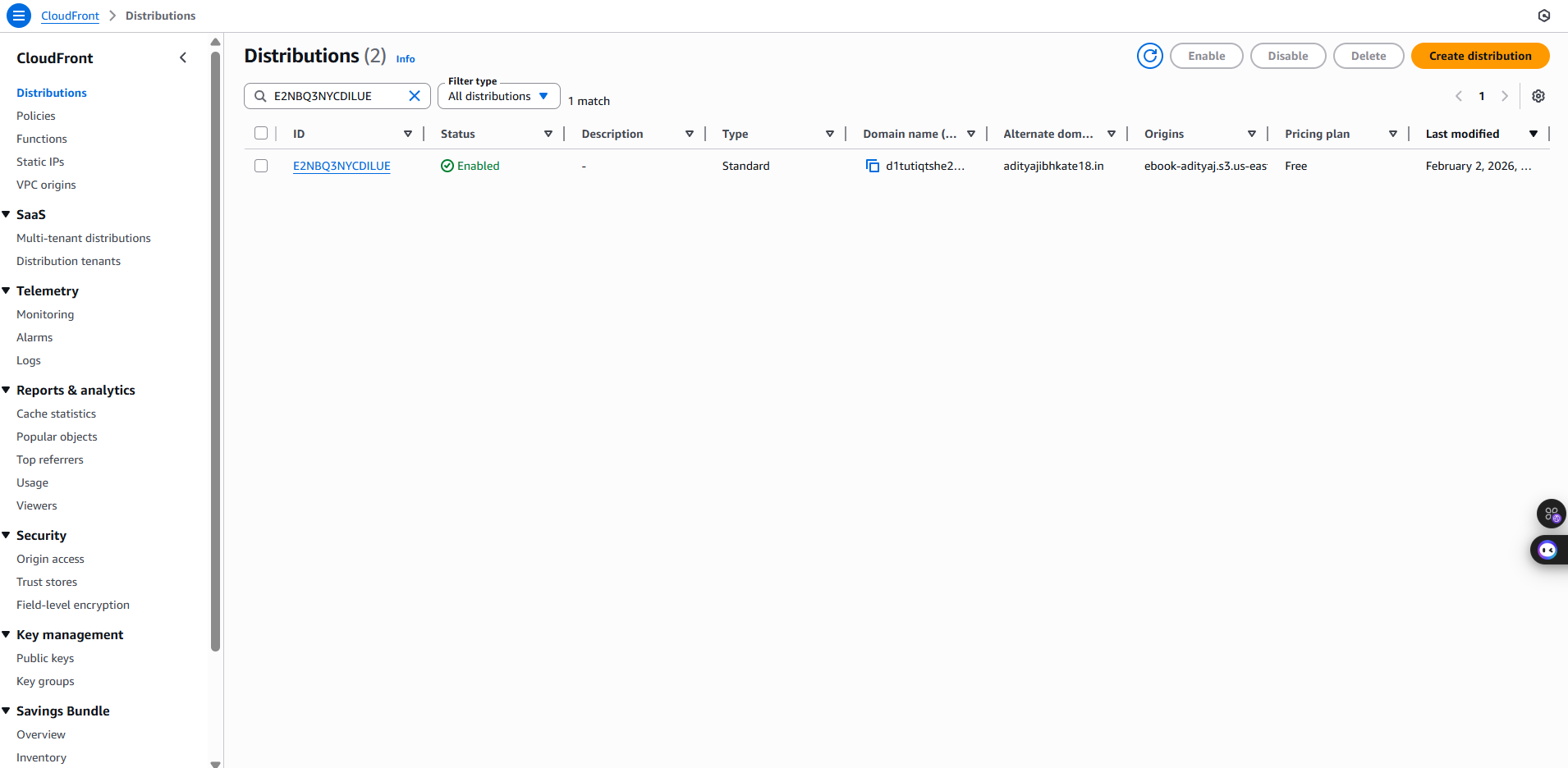
## 4. Static Website Hosting (Amazon S3)

• Created an S3 bucket to host frontend files.  
• Uploaded HTML, CSS, and JavaScript files.  
• Disabled public access.

  
• Configured bucket policy to allow access only from CloudFront.

## 5. Content Delivery Network (Amazon CloudFront)

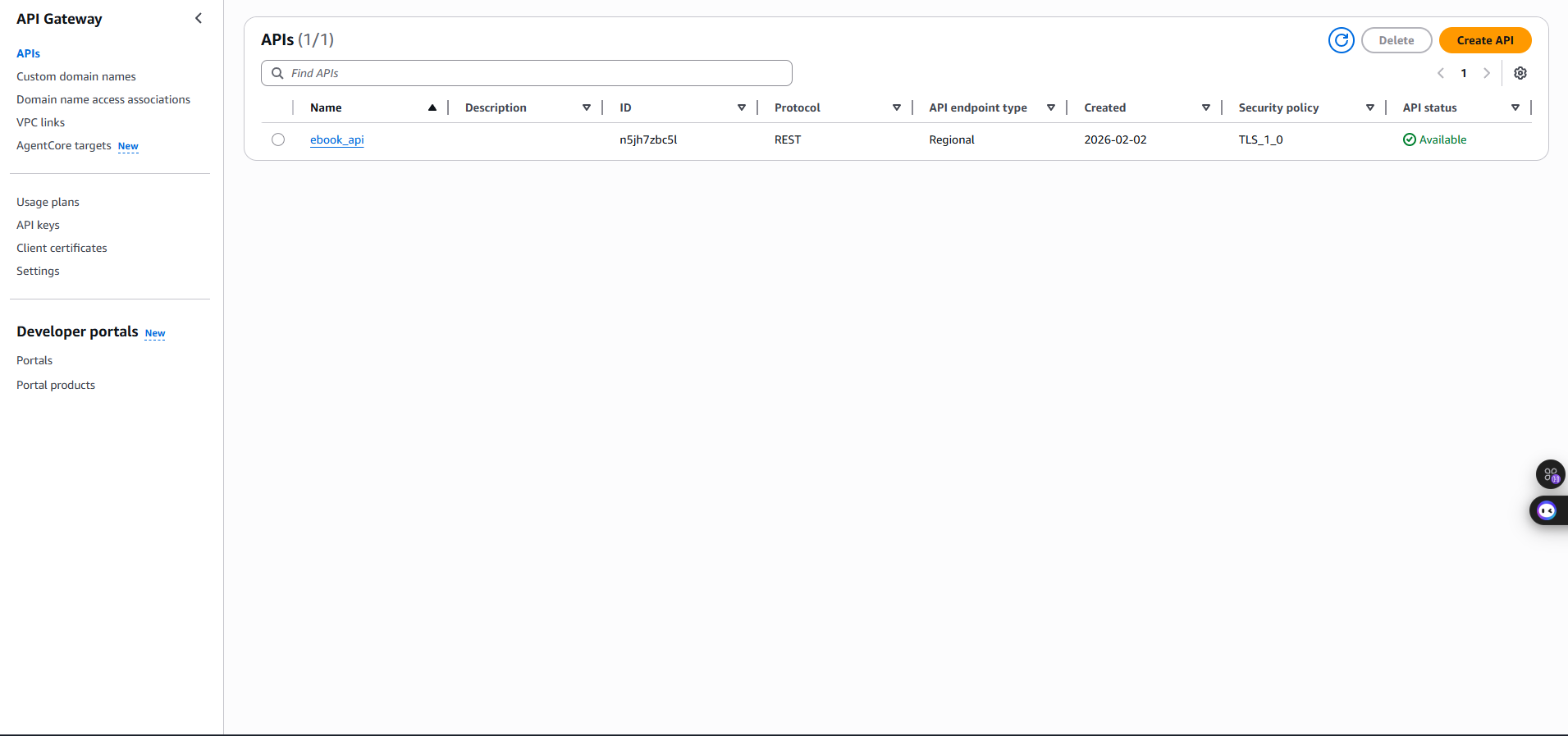
• Created a CloudFront distribution.



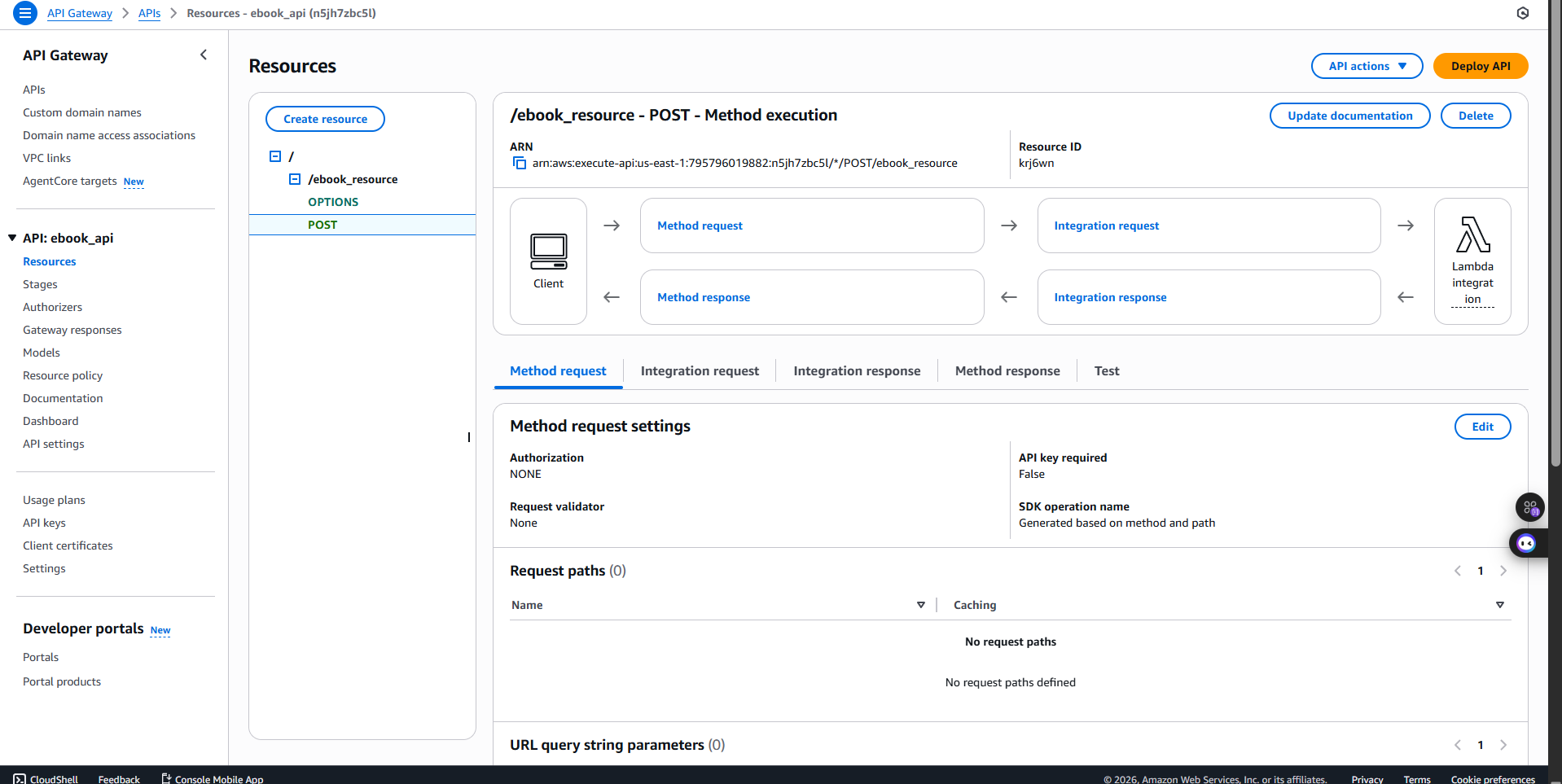
• Set S3 bucket as the origin.  
• Enabled HTTPS using ACM certificate.  
• Improved performance and security.

## 6. API Setup (Amazon API Gateway)

• Created REST API.

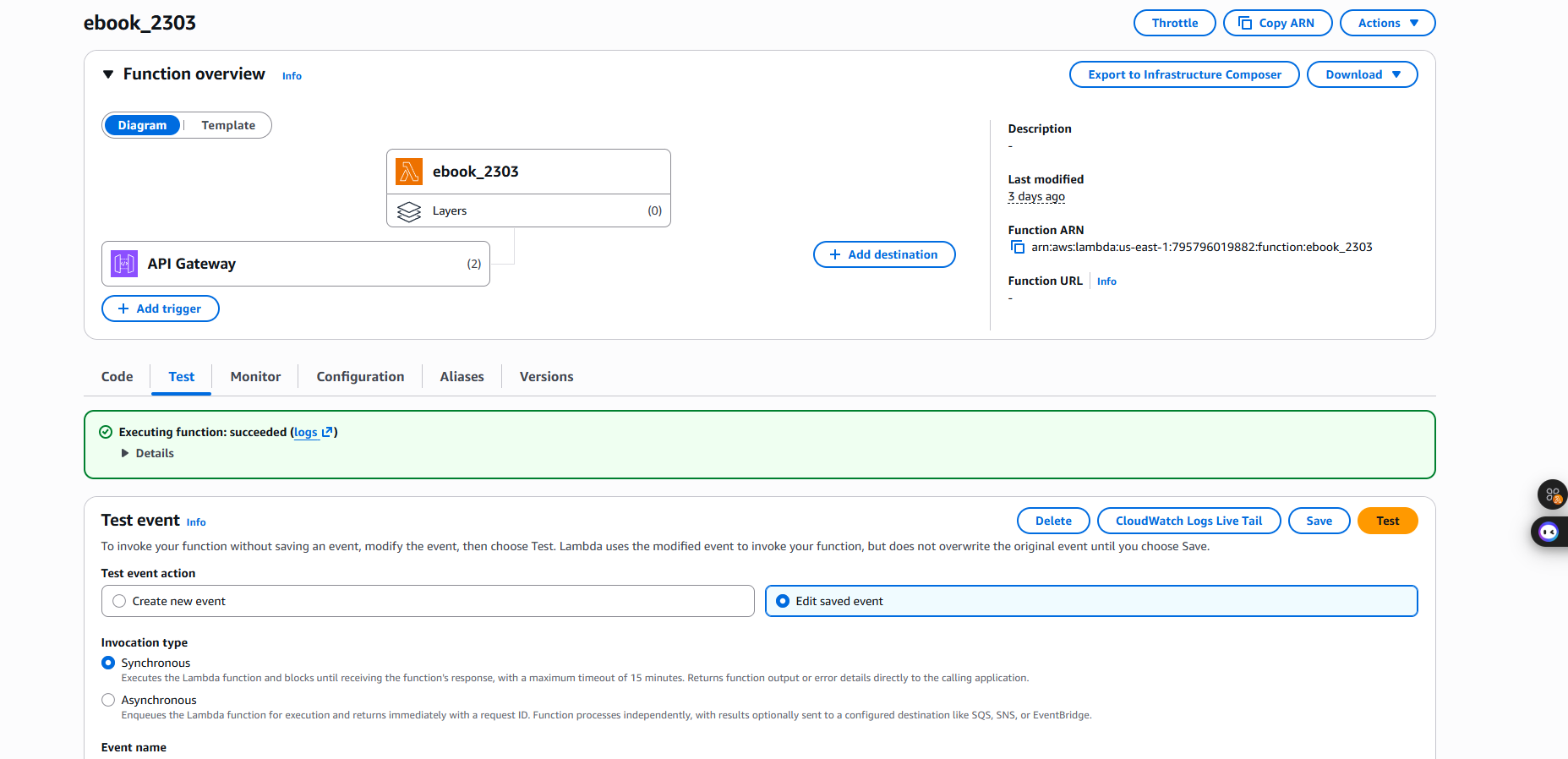
  
• Defined POST method for form submission.

• Integrated API with AWS Lambda.  
• Enabled CORS for frontend communication.



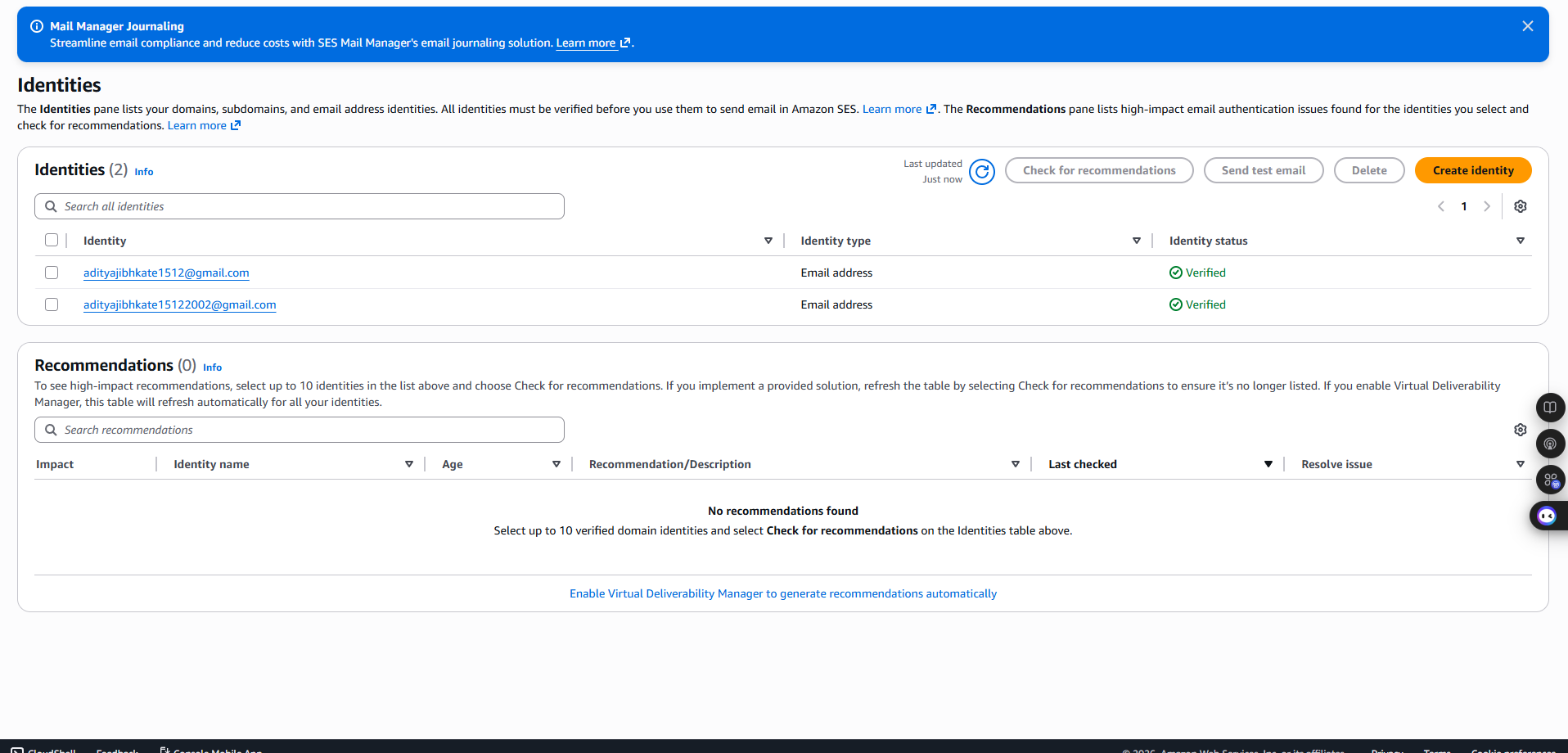
## 7. Backend Logic (AWS Lambda)

• Created Lambda function.  
• Added logic to validate form input.  
• Configured IAM role for SES and CloudWatch access.  
• Connected Lambda to API Gateway.



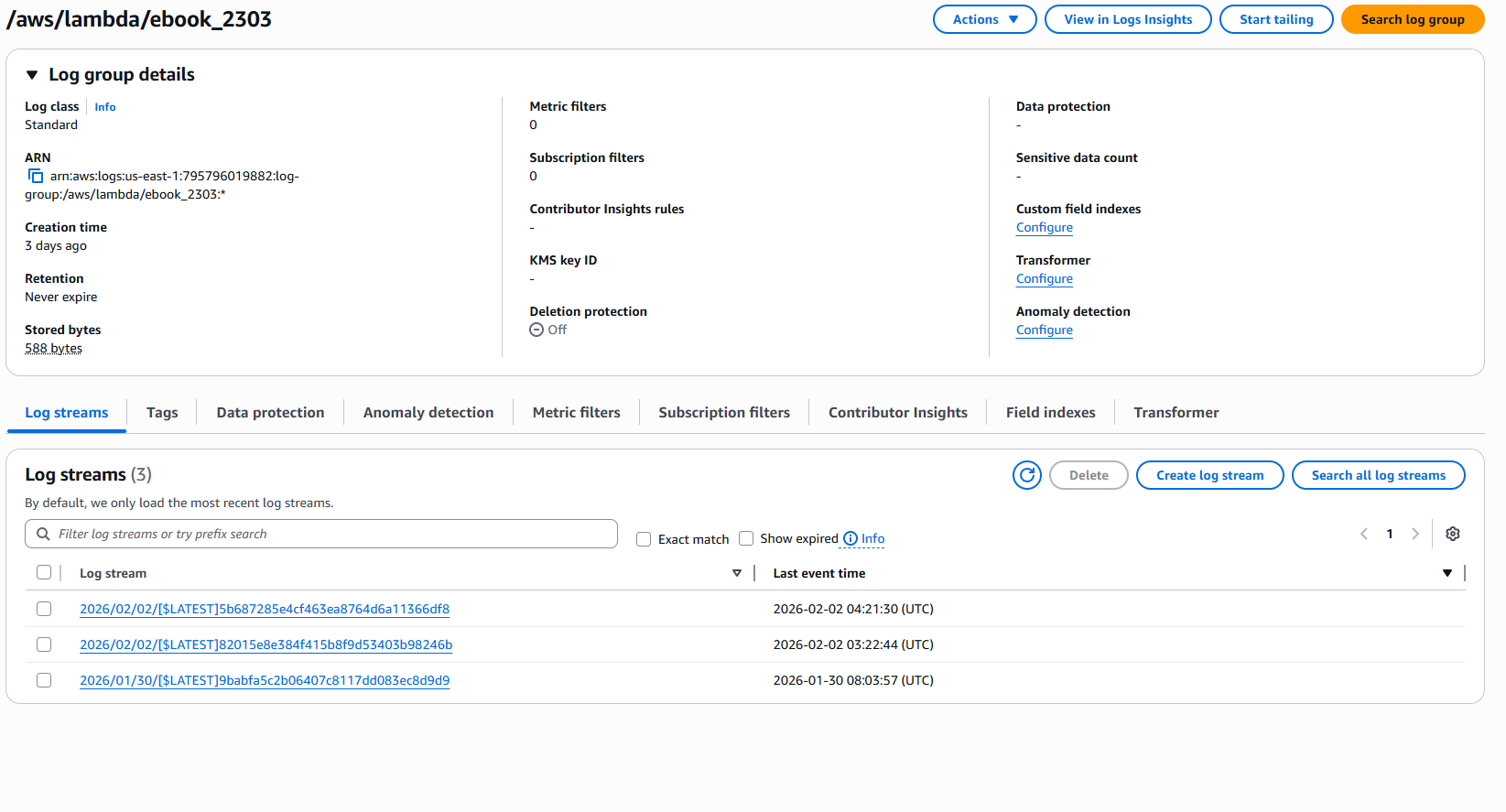
## 8. Email Service (Amazon SES)

• Verified email/domain identity.

  
• Configured SES sandbox or production access.  
• Enabled Lambda to send emails via SES.

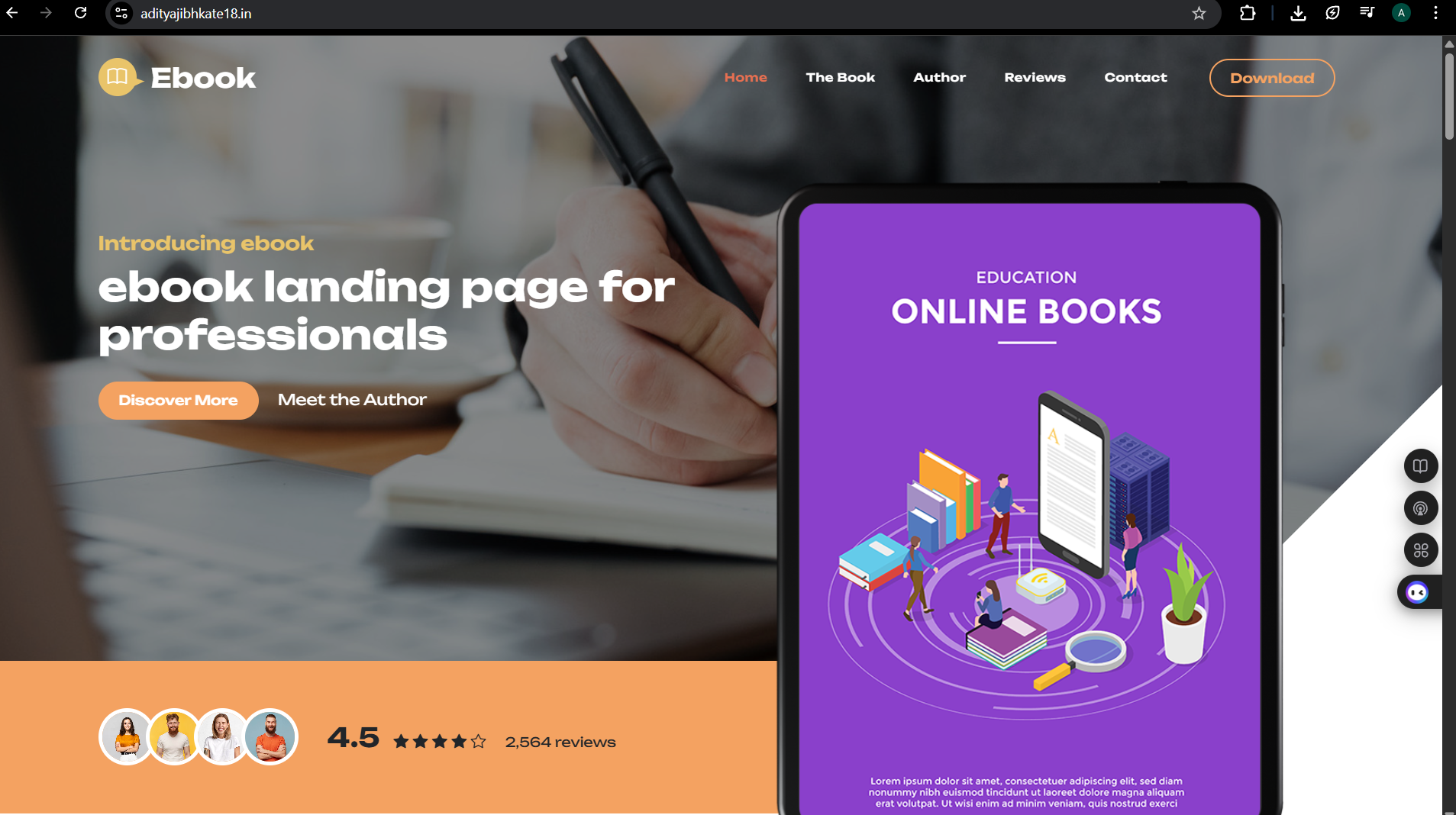
## 9. Monitoring & Logs (Amazon CloudWatch)

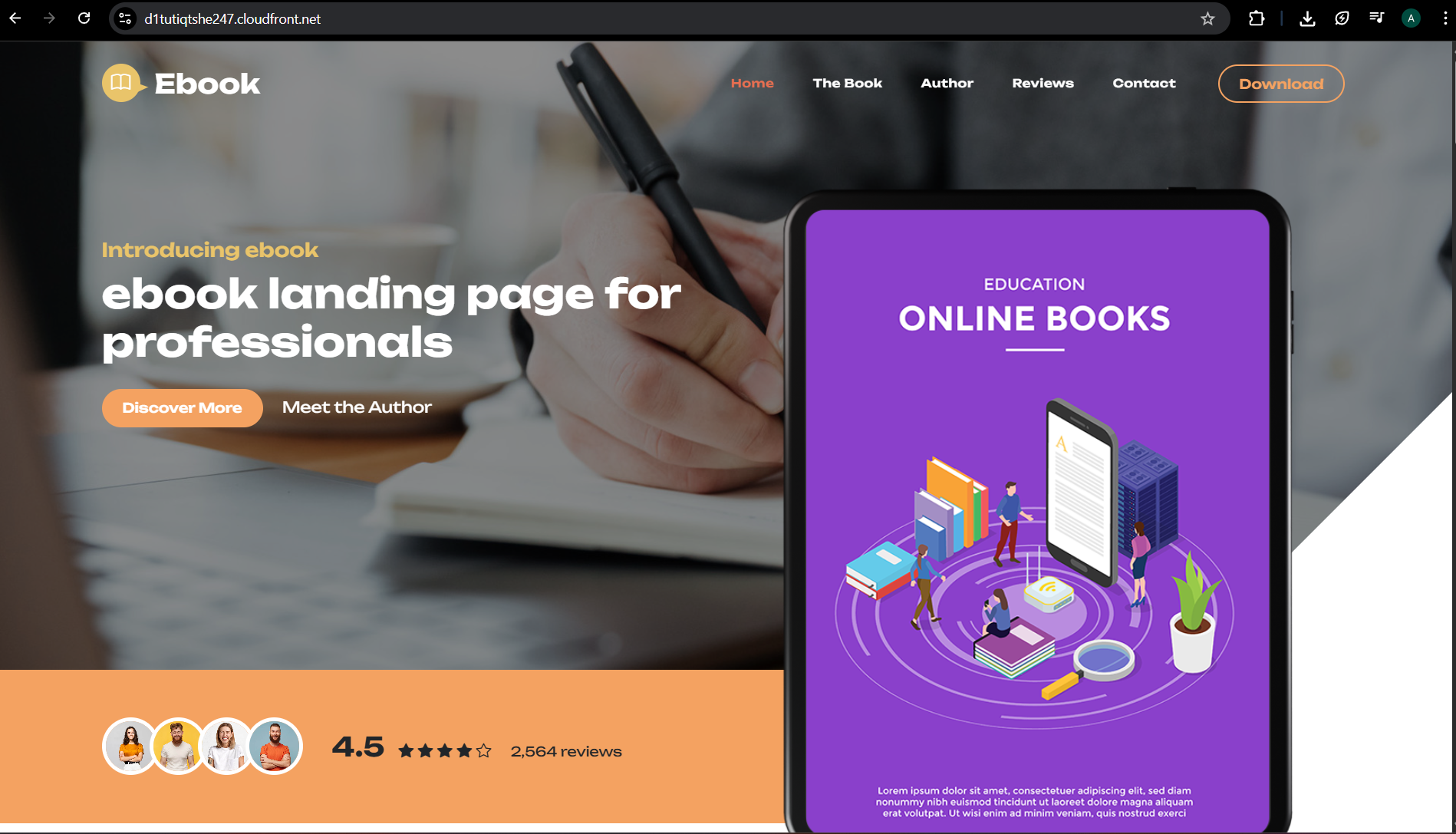
• Enabled CloudWatch logs for Lambda.  
• Monitored execution metrics.  
• Debugged errors using log streams.

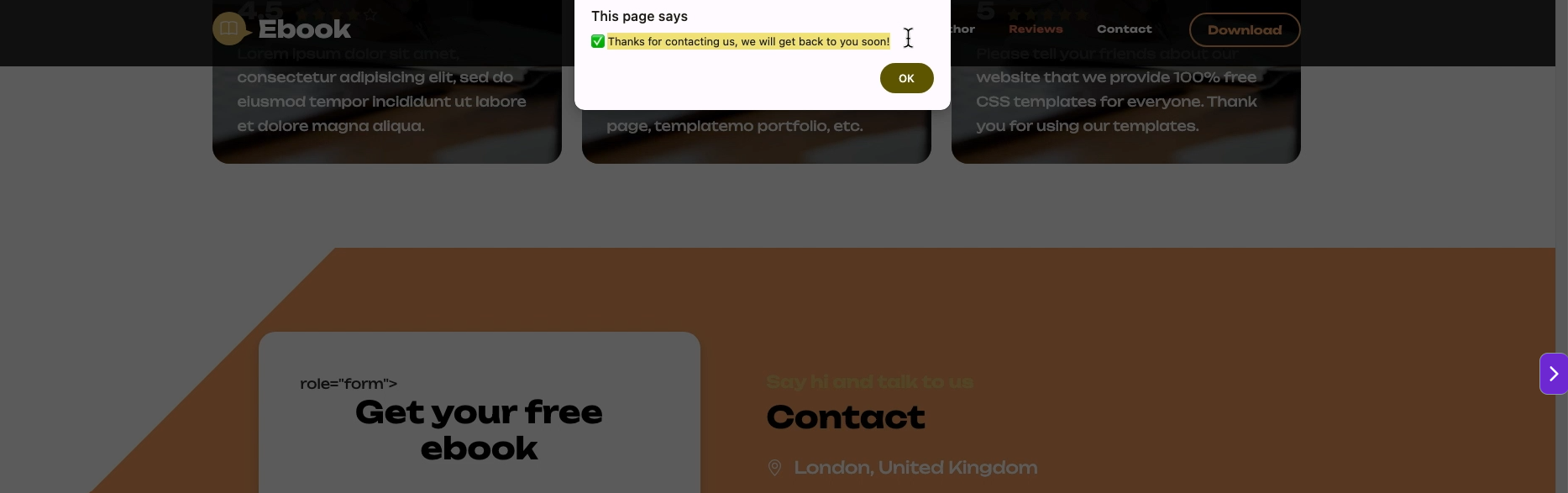


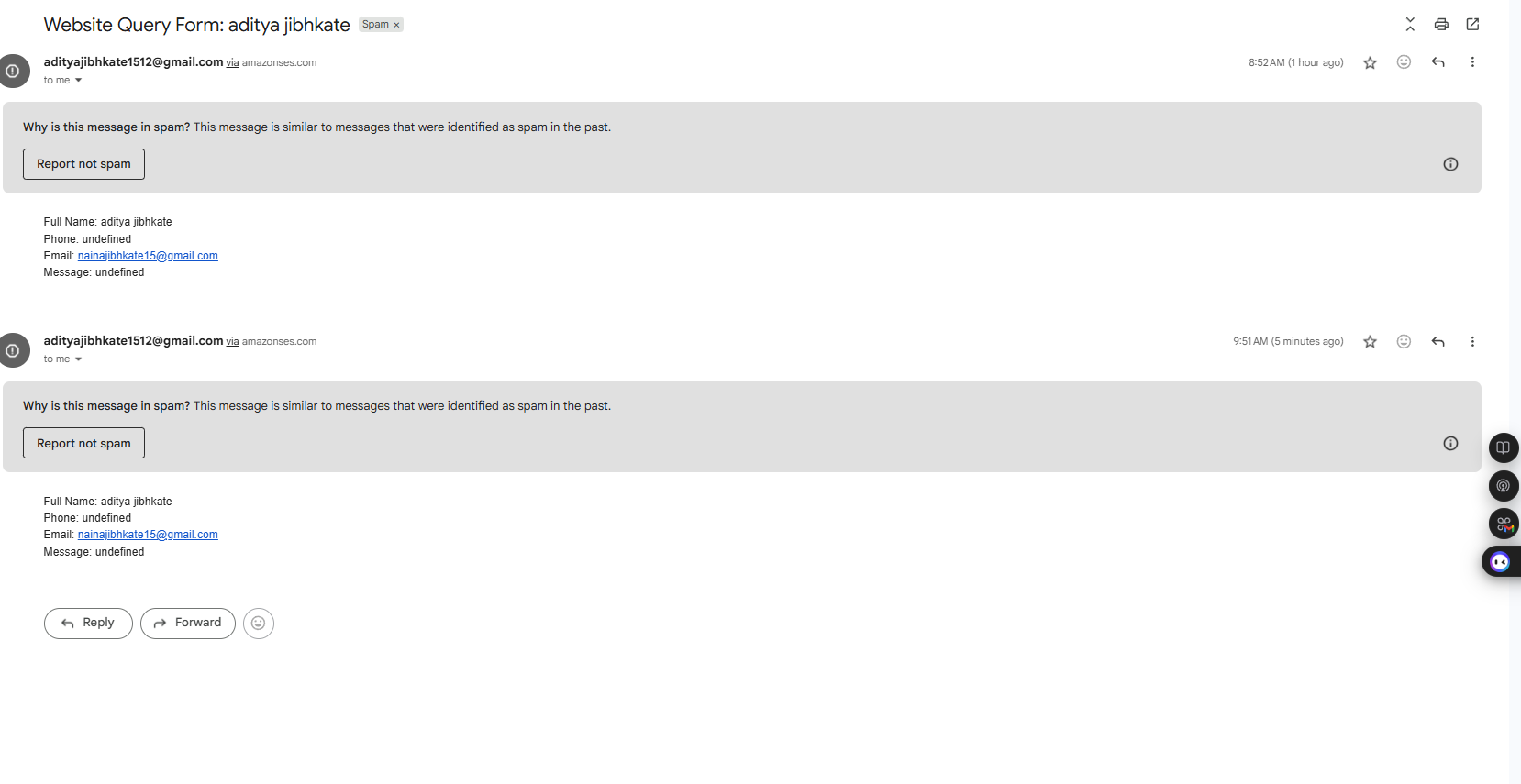
## 10. End-to-End Flow

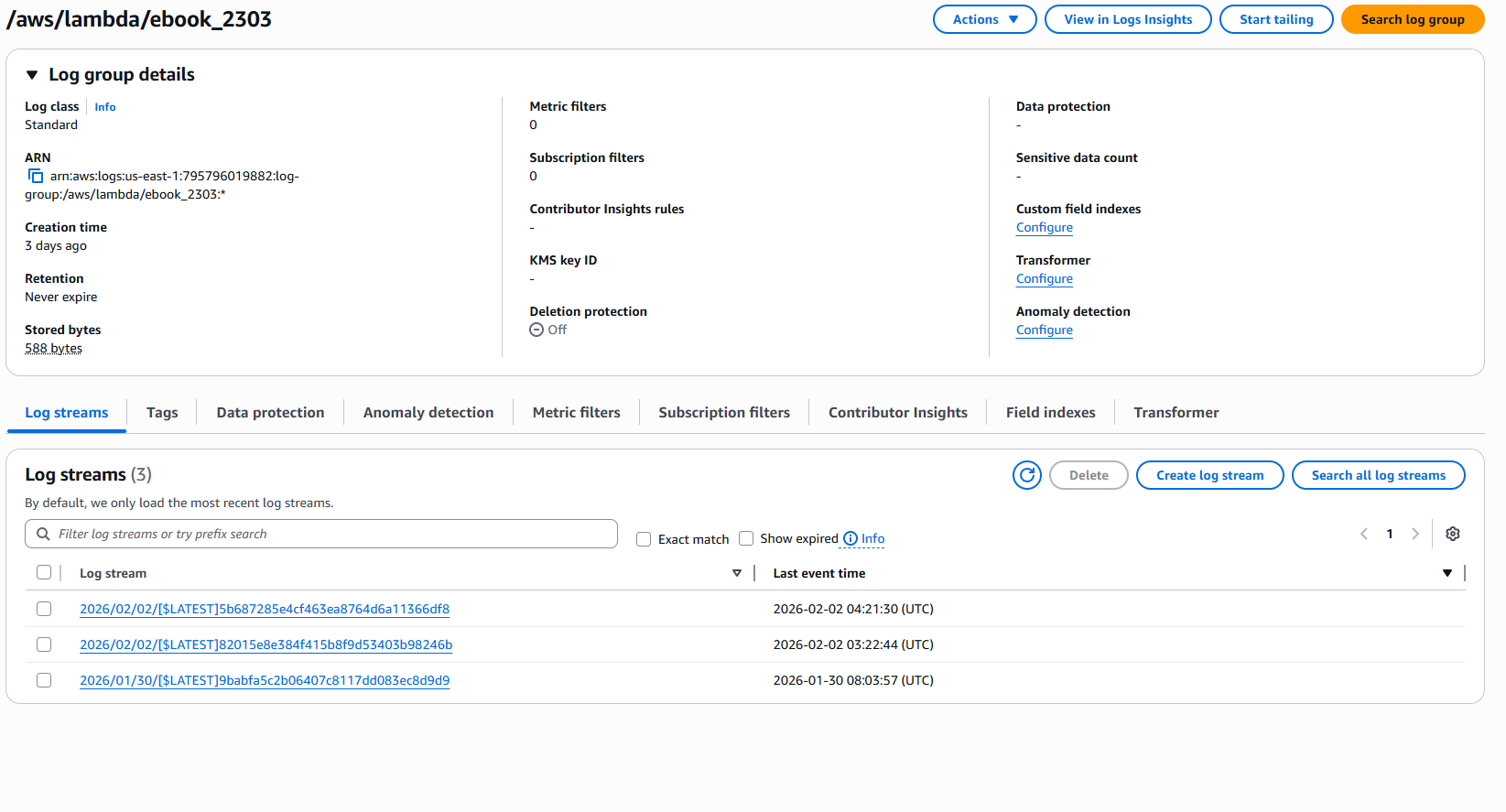
• User accesses website via domain.

  
• CloudFront serves static content.

  
• Form submission triggers API Gateway.

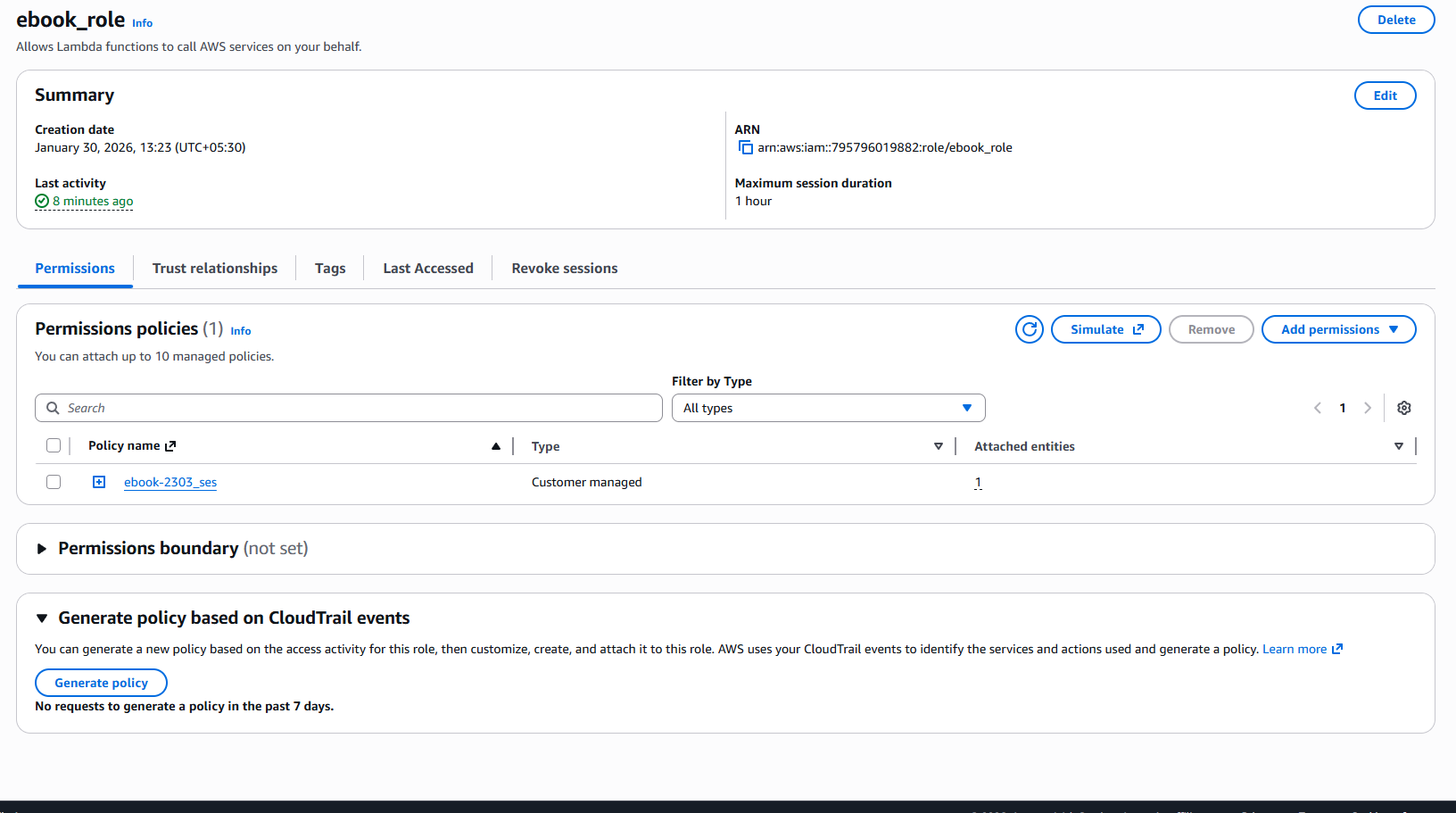
  
• Lambda processes request.  
• SES sends email notification.

  
• CloudWatch stores logs.



## 11. Security Best Practices

• Least privilege IAM roles.

  
• Private S3 bucket.  
• HTTPS enforced.  
• No hardcoded credentials.

## 12. Conclusion

This architecture provides a scalable, cost-effective, and secure serverless solution suitable for production environments.