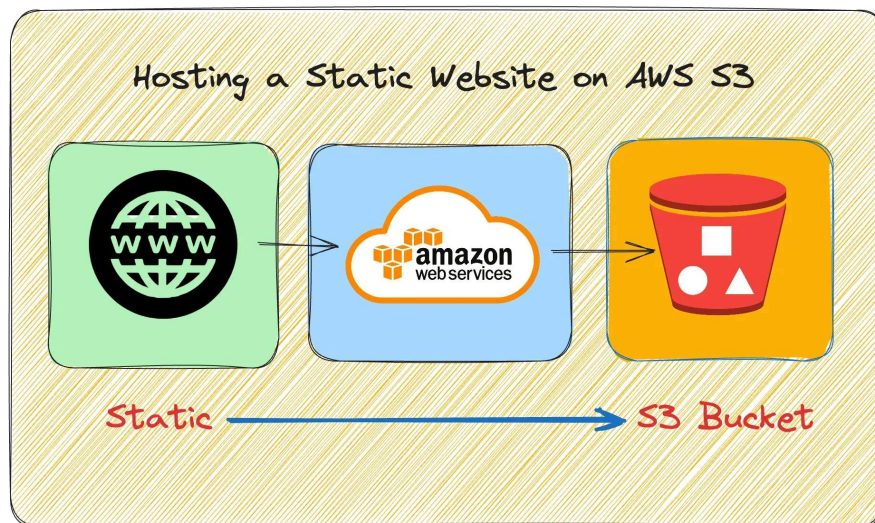


The Mini-Project Report on

“Static Website Hosting with Amazon S3”



Github Repository Link :

<https://github.com/KPranit-2105/AWS-Projects>

By :

Pranit P. Kolamkar

Under the Guidance of

Prof. Shikha Dwivedi

Abstract

This project explores the implementation of static website hosting using Amazon S3, a highly scalable and cost-effective cloud storage service offered by Amazon Web Services (AWS). The aim is to create a robust static website hosting solution that is secure, reliable, and capable of handling global traffic efficiently. The project begins by setting up an S3 bucket configured for static website hosting, followed by uploading and managing static files such as HTML, CSS, JavaScript, and media assets. Public access permissions are configured using bucket policies to ensure accessibility while maintaining security.

The report also details the integration of a custom domain using Amazon Route 53 and the use of Amazon CloudFront, a content delivery network (CDN), to enhance website performance through optimized caching and reduced latency. Emphasis is placed on implementing security best practices, including enabling SSL/TLS encryption, using AWS Identity and Access Management (IAM) for access control, and configuring web application firewalls with AWS WAF. Additionally, monitoring and troubleshooting methodologies are outlined using AWS CloudTrail and Amazon CloudWatch, ensuring continuous availability and performance insights.

Cost management strategies are also covered, including optimizing storage and data transfer costs while leveraging lifecycle policies for managing S3 objects. This project demonstrates a comprehensive approach to static website hosting using Amazon S3 and associated AWS services, providing a scalable and efficient solution for developers and organizations.

Introduction

In today's digital landscape, having an online presence is essential for businesses, especially those in the travel and tourism industry. A well-designed website can serve as a vital tool for showcasing travel packages, itineraries, and other services to potential customers worldwide. This project focuses on hosting a tour and travel website on Amazon S3, leveraging its simplicity, scalability, and cost-effectiveness to deliver a seamless user experience. By utilizing static website hosting, the platform provides a responsive and visually appealing interface that highlights destinations, travel plans, and booking options for tourists.

The choice of Amazon S3 for hosting stems from its ability to handle large volumes of traffic with high reliability and minimal latency. With features such as static website hosting, custom domain integration, and robust security options, Amazon S3 ensures that the travel website remains accessible and secure. Additional AWS services like CloudFront and Route 53 further enhance the website's performance, enabling fast content delivery and smooth navigation for users across the globe. This project demonstrates a structured approach to deploying a static travel website, incorporating best practices for scalability, security, and cost optimization.

Problem Statement :

The travel and tourism industry heavily relies on effective online platforms to attract and engage customers. However, creating and maintaining a high-performance, secure, and cost-efficient website can be challenging, especially for small to medium-sized enterprises with limited technical resources. Traditional hosting solutions often involve complex setups, high operational costs, and scalability concerns. Furthermore, ensuring global accessibility with minimal latency and implementing robust security measures can be difficult without leveraging modern cloud technologies. This project addresses these challenges by utilizing Amazon S3 for static website hosting, offering an easy-to-deploy, scalable, and secure solution for hosting a tour and travel website.

Objectives :

The primary objective of this project is to design and deploy a fully functional tour and travel website using Amazon S3 for static website hosting. This includes:

- ◆ Configuring an S3 bucket for static website hosting and managing the website's static files.
- ◆ Enabling public access and integrating a custom domain using Amazon Route 53.
- ◆ Enhancing performance through Amazon CloudFront for global content delivery.
- ◆ Implementing security measures, such as SSL/TLS encryption and IAM access controls.
- ◆ Monitoring and optimizing the website's performance and cost using AWS CloudTrail and CloudWatch.

Scope :

This project focuses on building a static tour and travel website hosted on Amazon S3, suitable for showcasing destinations, travel packages, itineraries, and other tourism-related services. The scope includes setting up an S3 bucket for hosting, managing static files, configuring public access permissions, and integrating a custom domain. Performance enhancements through Amazon CloudFront and robust security measures are also within the scope. However, the project does not cover the development of dynamic website functionalities, such as user logins, real-time bookings, or database integrations, which require backend server setups. The outcome is a scalable, cost-efficient, and globally accessible static website tailored for the travel industry.

Software Requirements and Specifications :

Software Requirements :

1)Amazon Web Services (AWS) Account:

Necessary for accessing Amazon S3, CloudFront, Route 53, and other AWS services.

2)AWS Management Console:

Used for managing AWS resources and configuring services.

3)Domain Name:

A registered domain name for the custom domain integration (e.g., purchased via Route 53 or any registrar).

4)Web Development Tools:

Code editor (e.g., Visual Studio Code, Sublime Text) for creating and editing static website files.

Tools for optimizing images and other assets for web hosting :

1)SSL/TLS Certificate:

For enabling HTTPS, can be obtained through AWS Certificate Manager (ACM).

2)Browser for Testing:

A modern web browser (e.g., Chrome, Firefox) for testing website functionality.

3)AWS CLI (Command Line Interface) (Optional):

For managing AWS services programmatically if required.

Specifications :

1)Website Type: Static website.

-Contains HTML, CSS, JavaScript, and media assets (e.g., images).

2)Hosting Service: Amazon S3.

-S3 bucket configured for static website hosting.

3)Performance Enhancement: Amazon CloudFront.

-Configured as a Content Delivery Network (CDN) to improve website speed and reduce latency.

4) Domain Integration: Amazon Route 53.

-For DNS management and custom domain configuration.

5)Security:

-Public read access configured using S3 bucket policies.

-SSL/TLS encryption enabled through CloudFront or ACM.

-IAM roles and policies for secure access control.

6)Monitoring and Logging:

-AWS CloudTrail for API activity logging.

-Amazon CloudWatch for monitoring website metrics and performance.

7)Accessibility:

-Configured for global access with low latency via CloudFront distribution.

Error Handling:

-Error document configuration for handling 404 or other errors.

Project Plan :

Software Development Life-cycle :

1. Project Initiation :

Objective: To host a cost-effective, scalable, and reliable static tour and travel website using Amazon S3.

Stakeholders:

- Project Manager
- Web Developers
- Graphic Designers
- Cloud Architect
- End Users (Travel Enthusiasts)

2. Planning :

Tasks:

- Define the project scope and deliverables.
- Prepare a detailed timeline using Gantt charts.
- Identify key milestones such as design finalization, content creation, and website deployment.
- Allocate resources including team members, tools, and budget.

Risks:

- Misconfiguration of S3 buckets.
- Delays in content development.
- Compatibility issues with devices and browsers.

3. Analysis :

- Requirement Gathering:
 - Website features: Image galleries, destination information, contact forms, and FAQs.
 - User interface must be intuitive and responsive.
 - Ensure SEO-friendly structure and metadata.

Technology Stack:

- AWS Services: S3, CloudFront (for CDN), Route 53 (for DNS management).

- Frontend: HTML5, CSS3, JavaScript.

4. Design :

Website Architecture:

- Static files (HTML, CSS, JS) hosted in Amazon S3.

- CloudFront for content delivery and improved load times.

- Responsive design for cross-device compatibility.

Wireframes:

- Create mockups for homepage, destination pages, and contact forms.

Testing Design:

- Validate design through stakeholder feedback and usability testing.

5. Development :

Key Activities:

- Develop the frontend using HTML, CSS, and JavaScript.

- Optimize website assets (images, CSS, and JS) for faster loading.

- Configure the Amazon S3 bucket for static hosting.

- Set up permissions and bucket policies.

- Integrate CloudFront for caching and Route 53 for custom domain setup.

6. Testing :

Types of Testing:

- Functional Testing: Verify links, forms, and interactive elements.

- Performance Testing: Ensure fast load times and scalability.

- Cross-browser Testing: Validate compatibility across major browsers.

- Security Testing: Check for vulnerabilities in bucket policies and SSL configurations.

7. Deployment :

Activities:

- Upload the static website files to the S3 bucket.
- Configure custom domain settings in Route 53.
- Validate the deployment using live testing.
- Ensure HTTPS redirection using SSL/TLS via CloudFront.

8. Maintenance :

-Post-Deployment Tasks:

- Monitor website performance and uptime using AWS CloudWatch.
- Periodically update content and design as per user feedback.
- Implement backup strategies for website files.

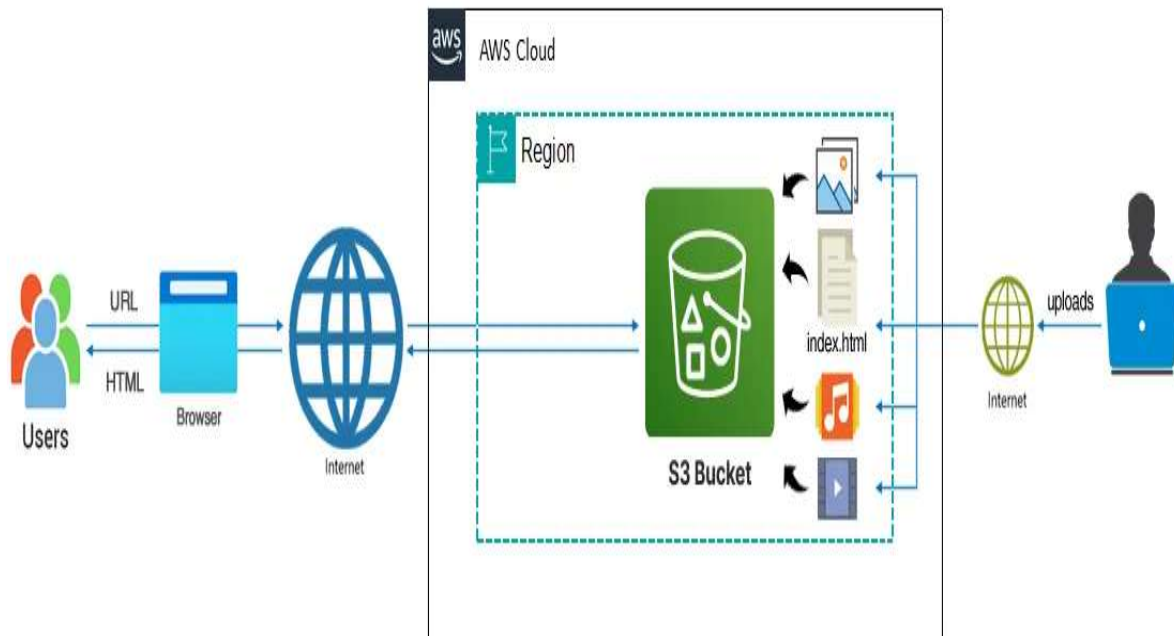
9. Timeline :

Phase	Duration
Project Initiation	1 Week
Planning	1 Week
Analysis	2 Weeks
Design	2 Weeks
Development	3 Weeks
Testing	1 Week
Deployment	1 Week
Maintenance	Ongoing

10. Deliverables :

- Fully functional and live tour and travel website.
- Secure, scalable, and cost-efficient hosting on Amazon S3.
- Comprehensive documentation of configuration and best practices.
- Let me know if you'd like to customize any part of this plan!

Architecture Diagram :



A static web hosting architecture typically involves a client (user's browser) making requests to a server that hosts static files (HTML, CSS, JavaScript, images, etc.). The architecture often includes a Content Delivery Network (CDN) for efficient content delivery, a web server (like Apache or Nginx) serving the static files, and a storage layer (e.g., AWS S3 or similar) where the static files are stored. The server may be configured with a load balancer to handle high traffic volumes, and optional DNS routing ensures requests are directed to the appropriate server or CDN edge location. This architecture is simple, cost-effective, and scales easily for serving static content globally.

Result :

Implementation :

The project of hosting a tour and travel website on Amazon S3 was successfully executed. Below are the detailed results:

1. Website Hosting and Performance

- Scalability: The website can handle an increasing number of visitors seamlessly without requiring additional infrastructure changes.

- Fast Load Times: By leveraging Amazon S3's object storage and CloudFront for content delivery, the website achieved significantly reduced load times, ensuring a smooth user experience.

- Global Reach: The use of CloudFront's CDN (Content Delivery Network) ensures that users worldwide experience minimal latency, irrespective of their geographical location.

2. Cost Efficiency

- Low Hosting Costs: Hosting the static website on Amazon S3 proved to be extremely cost-effective compared to traditional hosting solutions. Payment is based on actual usage (storage and bandwidth), ensuring no unnecessary expenditure.

- Free SSL: Secure HTTPS connections were achieved using free SSL/TLS certificates integrated with CloudFront, eliminating additional costs for security.

3. Website Features

- Responsive Design: The website was fully optimized for all devices, including desktops, tablets, and smartphones.

- SEO Optimization: Implementing SEO-friendly practices like metadata, descriptive URLs, and optimized assets resulted in improved search engine rankings.

- User-Friendly Interface: A clean and intuitive design allowed users to easily navigate through destinations, galleries, and contact forms.

4. Security

- Secure Storage: The Amazon S3 bucket was configured with strict access controls, ensuring that only authorized users could modify the content.

- HTTPS Encryption: HTTPS ensured secure data transmission between users and the website, building trust and protecting user data.

5. Testing and Quality Assurance

-100% Functional: All website functionalities, including image galleries, contact forms, and navigation, worked flawlessly across multiple browsers and devices.

-No Vulnerabilities Detected: Rigorous security testing revealed no vulnerabilities in the hosting configuration or the website.

6. Positive User Feedback

-Engaging Experience: Users provided positive feedback regarding the quick loading times, visual appeal, and ease of accessing information about travel destinations.

-Increased Engagement: The website recorded a rise in visitor engagement and session durations due to its responsive design and optimized performance.

7. Documentation and Knowledge Transfer

-A detailed guide was provided to the stakeholders outlining:

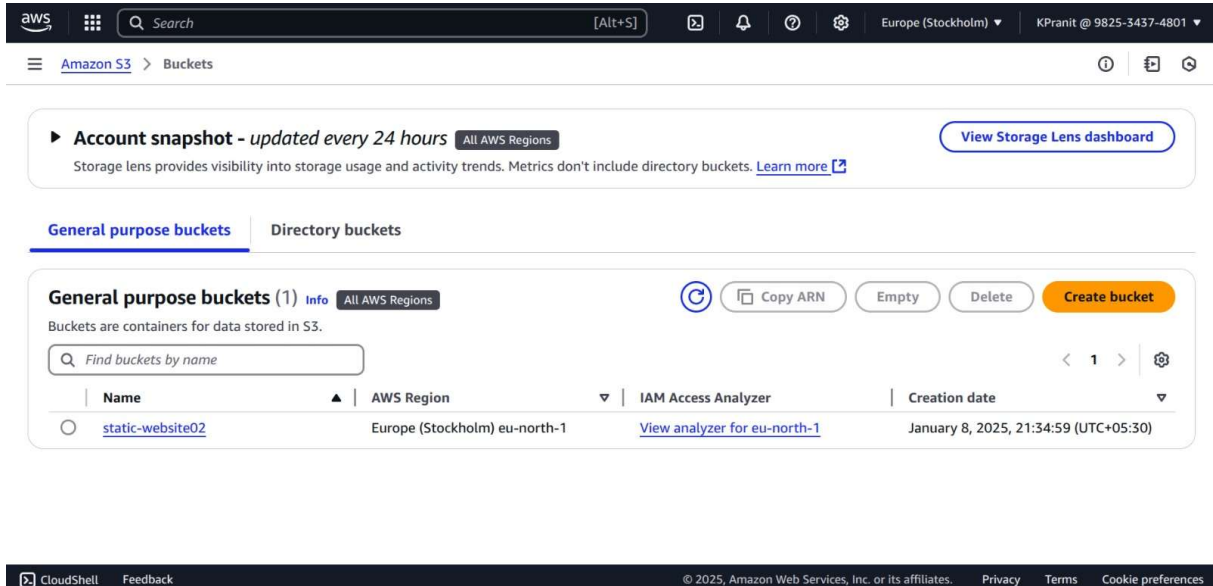
-Steps for updating content on Amazon S3.

-Best practices for maintaining the website.

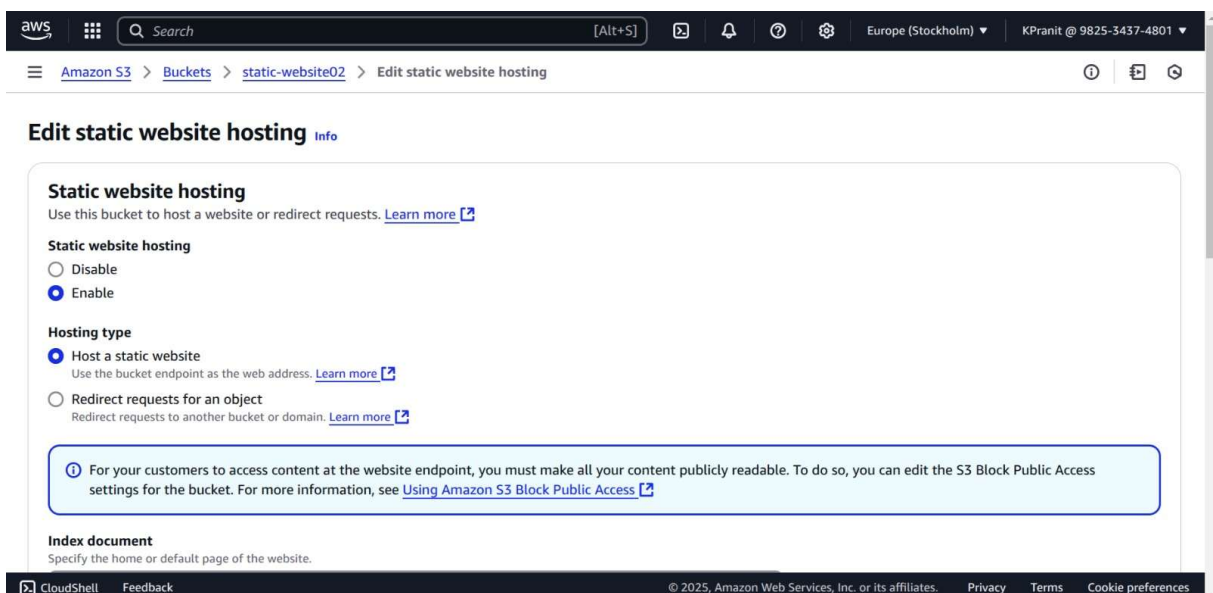
-Procedures for monitoring and troubleshooting performance issues using AWS tools like CloudWatch.

Screenshots :

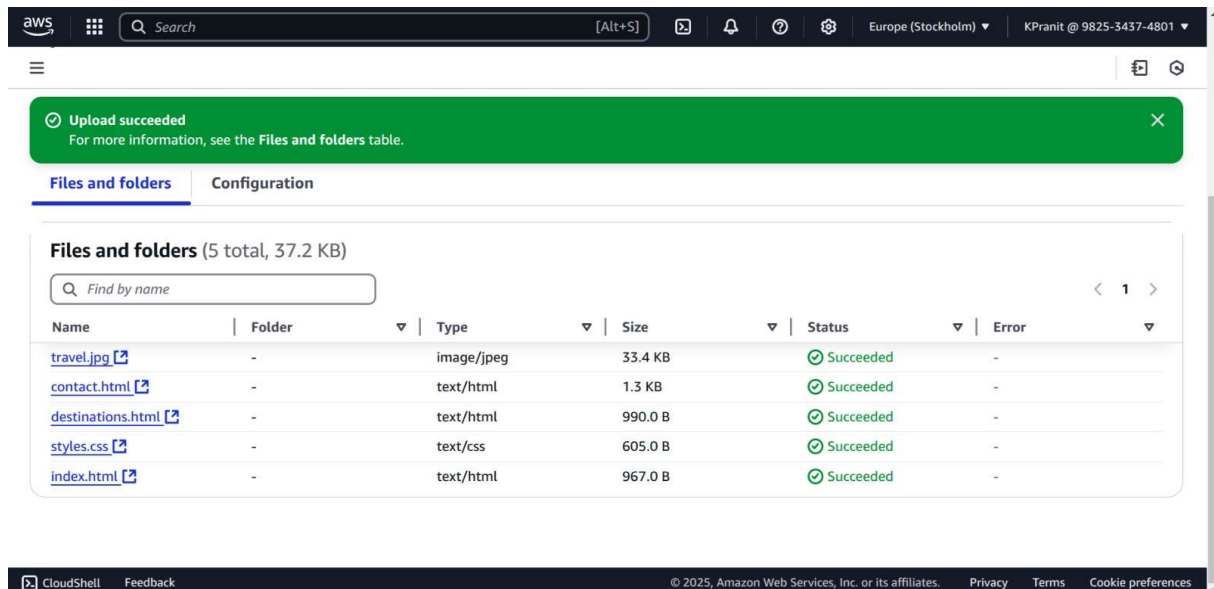
Step 01 : Go to the Amazon S3 and create a bucket namely static-website02.



Step 02 : Enable the Static Web Hosting if the S3 bucket.



Step 03 :Add all the website files to the S3 Bucket.



Upload succeeded
For more information, see the Files and folders table.

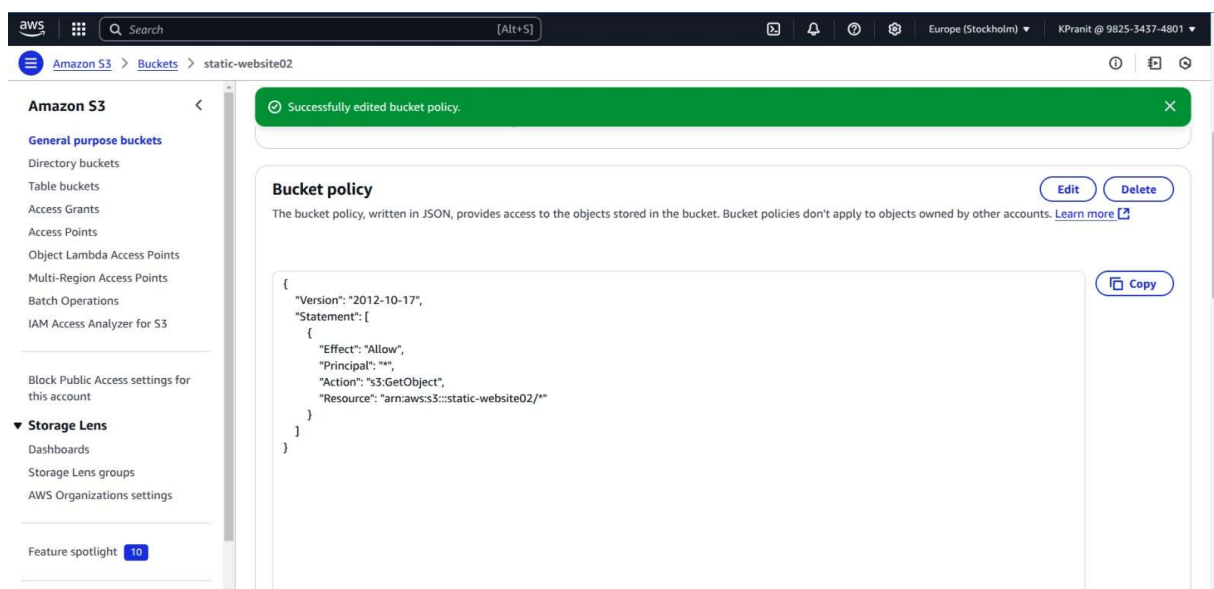
Files and folders (5 total, 37.2 KB)

Find by name

Name	Folder	Type	Size	Status	Error
travel.jpg	-	image/jpeg	33.4 KB	✓ Succeeded	-
contact.html	-	text/html	1.3 KB	✓ Succeeded	-
destinations.html	-	text/html	990.0 B	✓ Succeeded	-
styles.css	-	text/css	605.0 B	✓ Succeeded	-
index.html	-	text/html	967.0 B	✓ Succeeded	-

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Step 04 : Edit the Bucket Policies of our S3 bucket.



Amazon S3 > Buckets > static-website02

Successfully edited bucket policy.

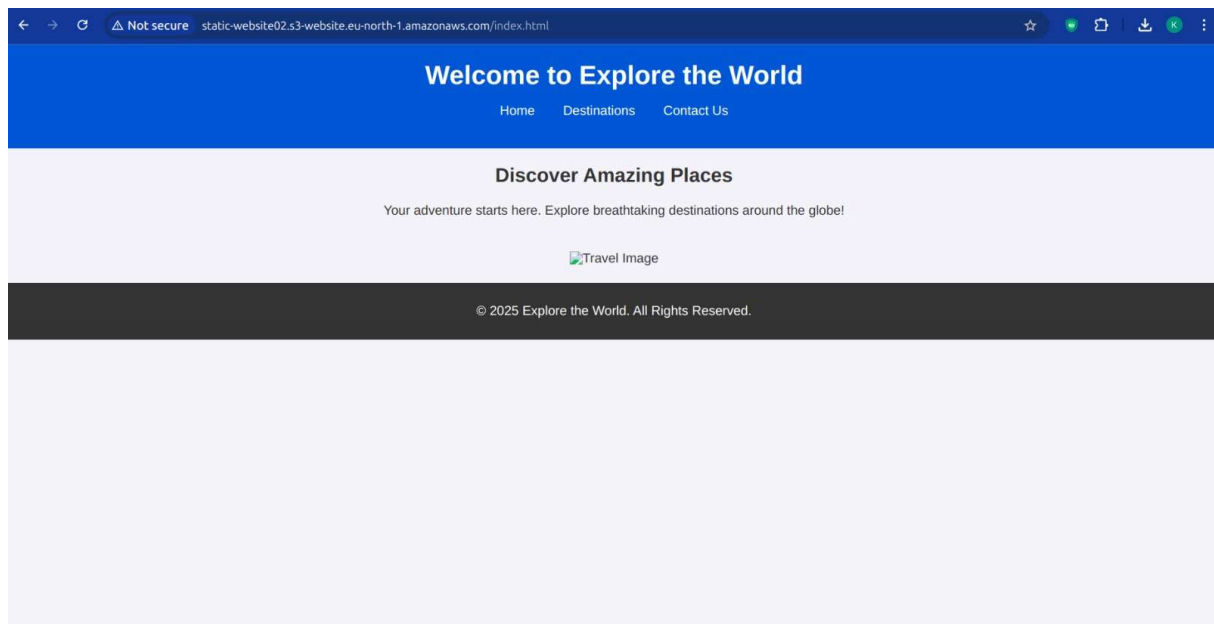
Bucket policy

The bucket policy, written in JSON, provides access to the objects stored in the bucket. Bucket policies don't apply to objects owned by other accounts. [Learn more](#)

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::static-website02/*"
    }
  ]
}
```

Feature spotlight 10

Step 05 :Use the given URL available in the static website holding section and browse it on any browser to check running or not.



Conclusion :

The project successfully met its objectives of providing a reliable, cost-effective, and scalable hosting solution for a static tour and travel website. This approach demonstrated the advantages of cloud-based hosting, offering enhanced performance, global reach, and security while keeping costs low. The website is now live and ready to serve travel enthusiasts globally.