

DEPLOYMENT OF STATIC WEBSITE USING AWS S3

PROJECT REPORT

Of

18CSC310J – Data Centric Networking and System Design

Submitted by

Yeluri Badrinath[RA2111028010111]

Madunuri Harshini[RA2111028010112]

Neha Bhardwaj[RA2111028010104]

Under the Guidance of

Dr. A.Helen Victoria

(Associate Professor, Networking and Communications)

*in partial fulfillment of the requirements for the degree
of*

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE ENGINEERING

with specialization in (Cloud Computing)



**DEPARTMENT OF NETWORKING AND COMMUNICATIONS
COLLEGE OF ENGINEERING AND TECHNOLOGY
SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
KATTANKULATHUR- 603 203**

NOVEMBER 2023

INDEX

S.NO	CONTENT	PAGE NO
1.	Abstract	3
2.	Introduction	4
3.	Motivation	5
4.	Challenges	6
5.	Architecture Diagram	8
6.	Working Model	9
7.	Results	10
8.	Conclusion	13
9.	References	14

ABSTRACT

This abstract outlines the process of deploying a static website using Amazon Simple Storage Service (S3) on the Amazon Web Services (AWS) platform. It covers key components such as creating an S3 bucket, uploading static content, configuring domains, implementing SSL/TLS security, and optimizing content delivery through Amazon CloudFront. The abstract emphasizes the benefits of cost-effectiveness, scalability, and integration with other AWS services, providing a foundation for a comprehensive guide on the topic.

INTRODUCTION

In the ever-evolving landscape of web development, the deployment of static websites has emerged as a foundational practice for delivering digital content seamlessly and efficiently. As the demand for robust, scalable, and cost-effective solutions continues to rise, Amazon Web Services (AWS) offers a compelling platform for hosting static websites through its Simple Storage Service (S3).

Historically, web hosting has undergone transformative shifts, with static websites regaining prominence for their simplicity and performance advantages. AWS S3, a highly reliable and scalable object storage service, has become a cornerstone in this resurgence, providing developers with a versatile solution for hosting static web content.

This introduction sets the stage for exploring the intricacies of deploying a static website using AWS S3. As we navigate through the key components of this deployment process, from the creation of S3 buckets to the optimization of content delivery through Amazon CloudFront, we will uncover the simplicity, scalability, and economic advantages that make this approach a compelling choice for developers and businesses alike.

Join us on this journey as we delve into the comprehensive deployment steps, unlocking the potential of AWS S3 to streamline the hosting of static websites and empower developers to deliver content that is not only visually compelling but also highly performant and securely accessible.

MOTIVATION

Motivation for Deployment of a Static Website using AWS S3:

The motivation behind deploying a static website using Amazon Simple Storage Service (S3) on the Amazon Web Services (AWS) platform lies in a convergence of factors that address the contemporary needs of web developers, businesses, and end-users alike.

1. Simplicity and Speed:

- Static websites are inherently simpler, composed of fixed HTML, CSS, and JavaScript files. This simplicity translates into quicker load times, providing a seamless and responsive user experience.

2. Cost-Effectiveness:

- AWS S3 operates on a pay-as-you-go model, making it an economical choice for hosting static websites.

3. Scalability and Reliability:

- AWS S3's scalable architecture allows websites to handle varying levels of traffic effortlessly. Whether experiencing a surge in visitors or steady day-to-day traffic, the infrastructure adapts dynamically, ensuring consistent performance and reliability.

4. Security and Compliance:

- The integration of Amazon S3 with AWS Identity and Access Management (IAM) enables precise control over access permissions.

5. Content Delivery Optimization:

- Amazon CloudFront, when paired with AWS S3, facilitates content delivery optimization. By distributing content across a global network of edge locations, users experience reduced latency, faster load times, and enhanced overall performance.

6. Developer Flexibility:

- AWS S3 accommodates various development workflows, allowing developers to deploy and manage static website content using their preferred tools and frameworks. This flexibility fosters innovation and encourages the use of diverse technologies within the AWS ecosystem.

CHALLENGES

While deploying a static website using Amazon Simple Storage Service (S3) offers numerous benefits, there are specific challenges that developers may encounter during the implementation process. Understanding and addressing these challenges is crucial for ensuring a seamless deployment. Here are some key challenges associated with deploying a static website using AWS S3:

1. Limited Dynamic Functionality:

- Static websites are inherently limited in dynamic functionality compared to dynamic websites. Implementing interactive features, server-side logic, or real-time data updates can be challenging without resorting to additional services or hybrid architectures.

2. Workflow Complexity for Continuous Deployment:

- Establishing an efficient workflow for continuous deployment and version control can be challenging. Integrating S3 deployment into a robust CI/CD (Continuous Integration/Continuous Deployment) pipeline requires careful configuration to avoid disruptions in service during updates.

3. Fine-Grained Security Controls:

- While AWS S3 offers robust security features, configuring fine-grained access controls and permissions might be challenging for users new to AWS IAM (Identity and Access Management). Ensuring the right balance between security and accessibility requires careful consideration.

4. Search Engine Optimization (SEO) Challenges:

- Static websites may face SEO challenges, especially in scenarios where frequent content updates are required. Implementing SEO best practices, such as dynamic meta tags or server-side rendering for certain content, may be more complex in a purely static environment.

5. Dependency Management and External APIs:

- Managing dependencies and integrating with external APIs can be more challenging in a static website deployment. While client-side technologies can handle some dynamic interactions, certain functionalities may necessitate additional server-side processing.

6. Large File Handling:

- Handling large files, such as high-resolution images or videos, can present challenges in terms of storage costs and data transfer fees. Optimizing and compressing assets are critical for maintaining cost efficiency and ensuring optimal website performance.

7. Error Handling and Redirects:

- Configuring error handling and redirects requires careful attention to detail. Custom error pages and redirects need to be defined accurately in both the AWS S3 bucket settings and potentially in a content delivery network (CDN) like Amazon CloudFront.

8. Understanding and Optimizing Costs:

- While AWS S3 is cost-effective, understanding the pricing model and optimizing costs can be a challenge. Configuring lifecycle policies, setting up proper logging, and monitoring usage are essential to avoid unexpected charges.

9. Cross-Origin Resource Sharing (CORS) Configuration:

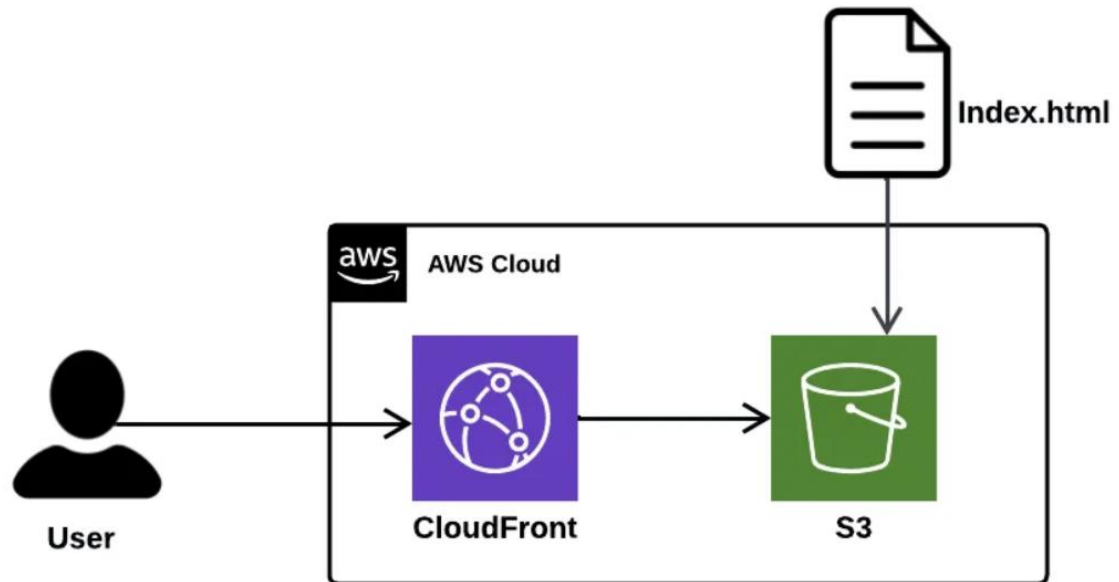
- Managing CORS configurations is crucial, especially when integrating with third-party services or APIs. Ensuring that the necessary cross-origin permissions are correctly configured in the S3 bucket can be challenging for those unfamiliar with CORS.

10. Global Latency and Performance Considerations:

- Although Amazon CloudFront can significantly enhance global performance, configuring it optimally and understanding the impact of edge locations on latency requires careful consideration and testing.

Addressing these challenges requires a combination of careful planning, configuration expertise, and a solid understanding of the specific requirements of the static website being deployed. As AWS services continue to evolve, new tools and features may emerge to simplify or address some of these challenges.

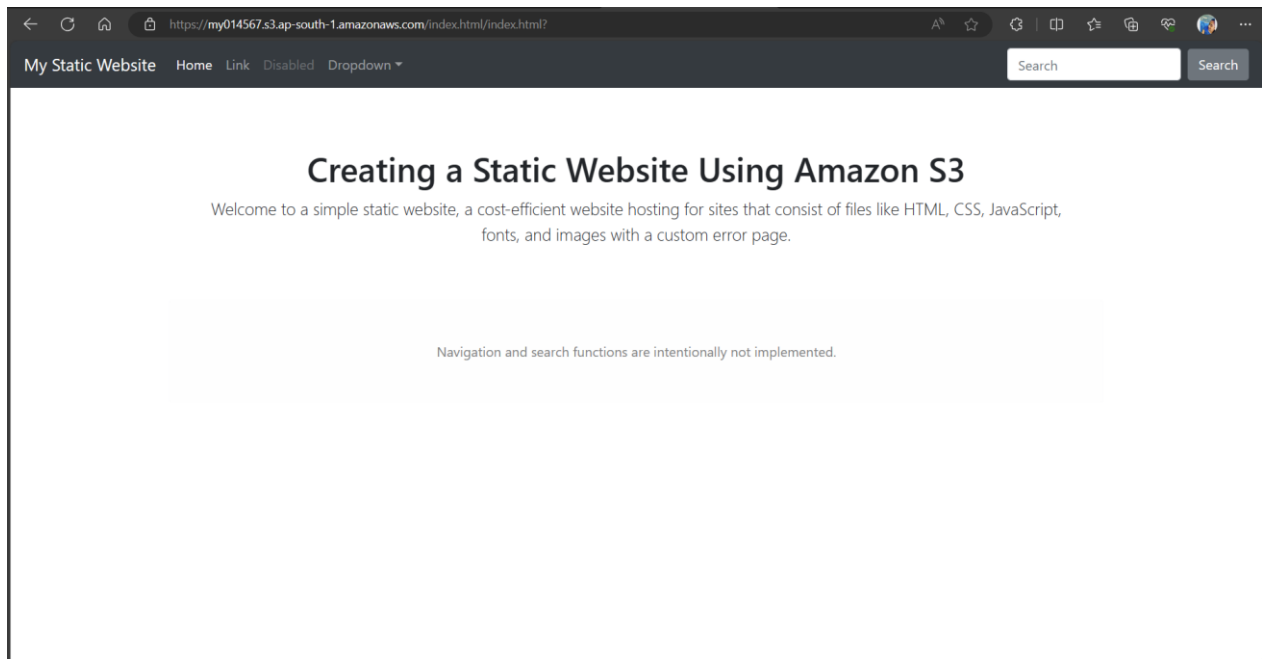
ARCHITECTURE DIAGRAM



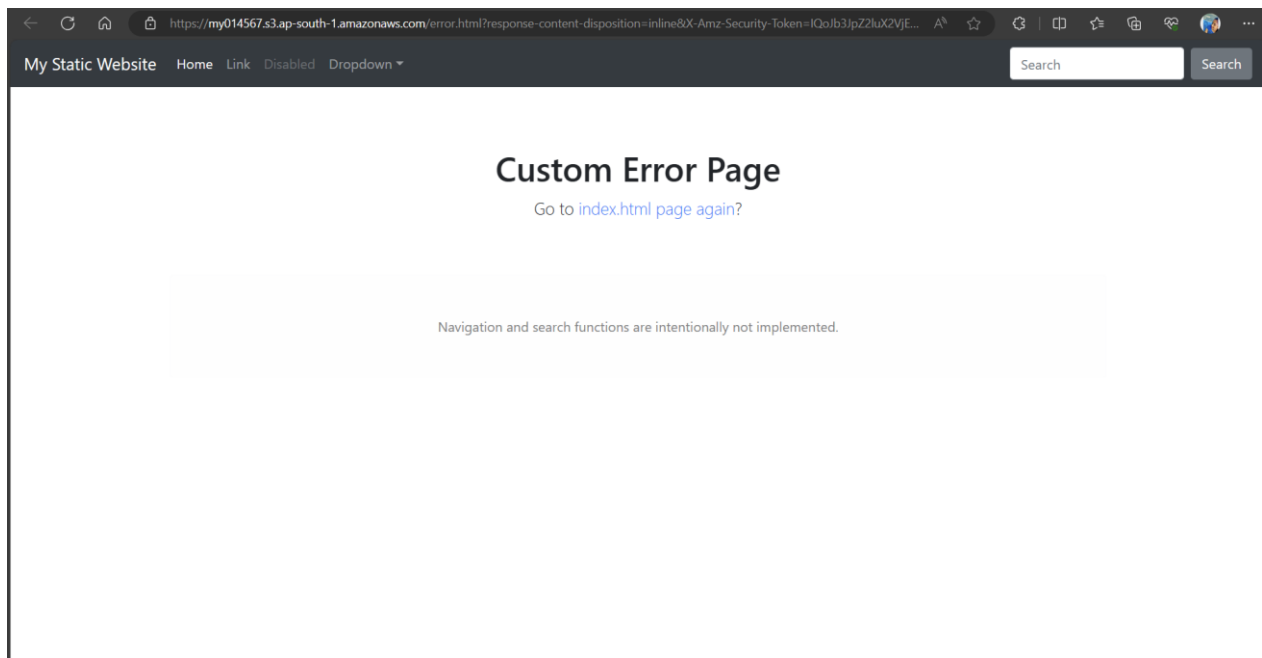
- ARCHITECTURE DIAGRAM FOR DEPLOYMENT OF STATIC WEBSITE USING AWS S3

WORKING MODEL

FOR INDEX.HTML



FOR ERROR.HTML



Results

The screenshot shows the Amazon S3 Buckets console. The left sidebar contains navigation links for Buckets, Access Points, Object Lambda Access Points, Multi-Region Access Points, Batch Operations, IAM Access Analyzer for S3, Storage Lens, Dashboards, Storage Lens groups, and AWS Organizations settings. The main content area displays an 'Account snapshot' and a list of 5 buckets. The buckets are:

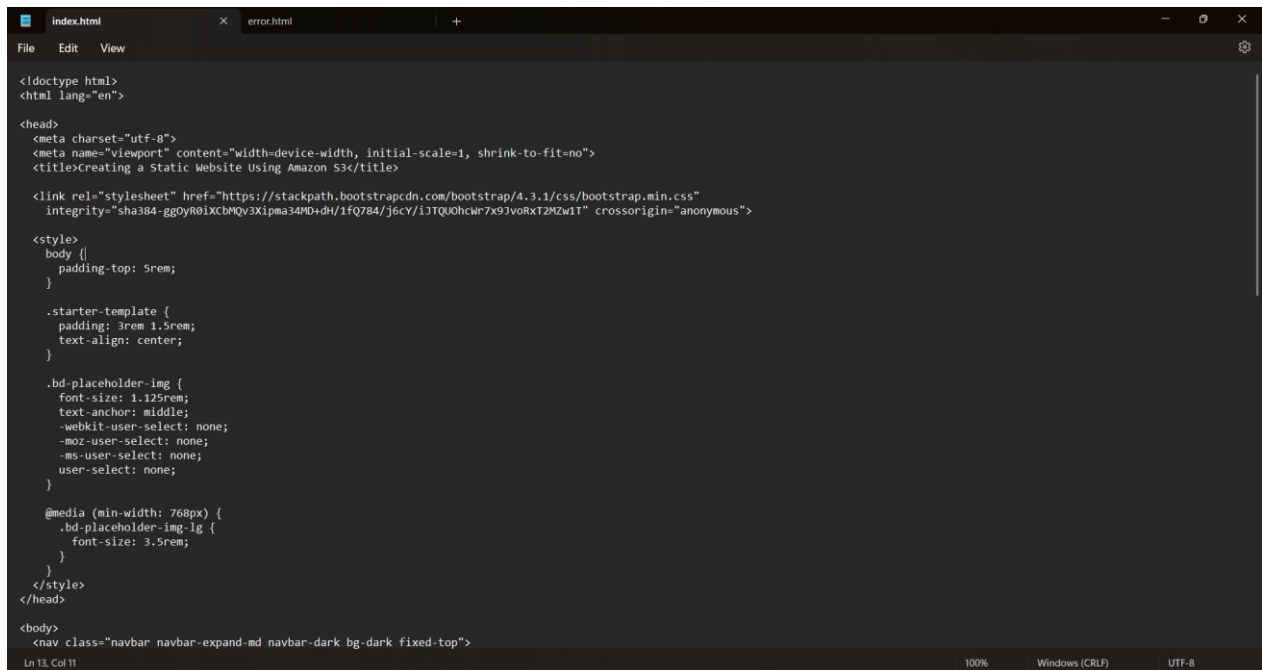
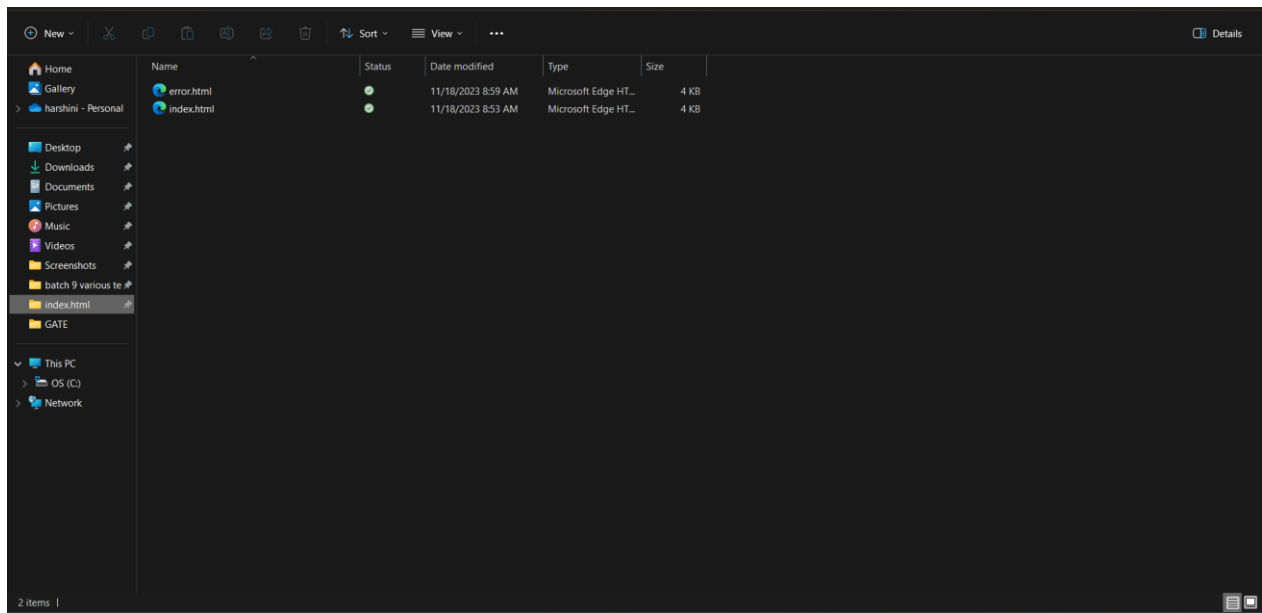
Name	AWS Region	Access	Creation date
dolly-new	US East (N. Virginia) us-east-1	Public	August 28, 2023, 10:16:23 (UTC+05:30)
harshininew	Asia Pacific (Sydney) ap-southeast-2	Bucket and objects not public	September 6, 2023, 10:18:09 (UTC+05:30)
idk-new	Asia Pacific (Mumbai) ap-south-1	Bucket and objects not public	September 19, 2023, 10:03:53 (UTC+05:30)
my014567	Asia Pacific (Mumbai) ap-south-1	Public	November 15, 2023, 21:55:15 (UTC+05:30)
vijay-new	Asia Pacific (Mumbai) ap-south-1	Bucket and objects not public	September 19, 2023, 10:03:00 (UTC+05:30)

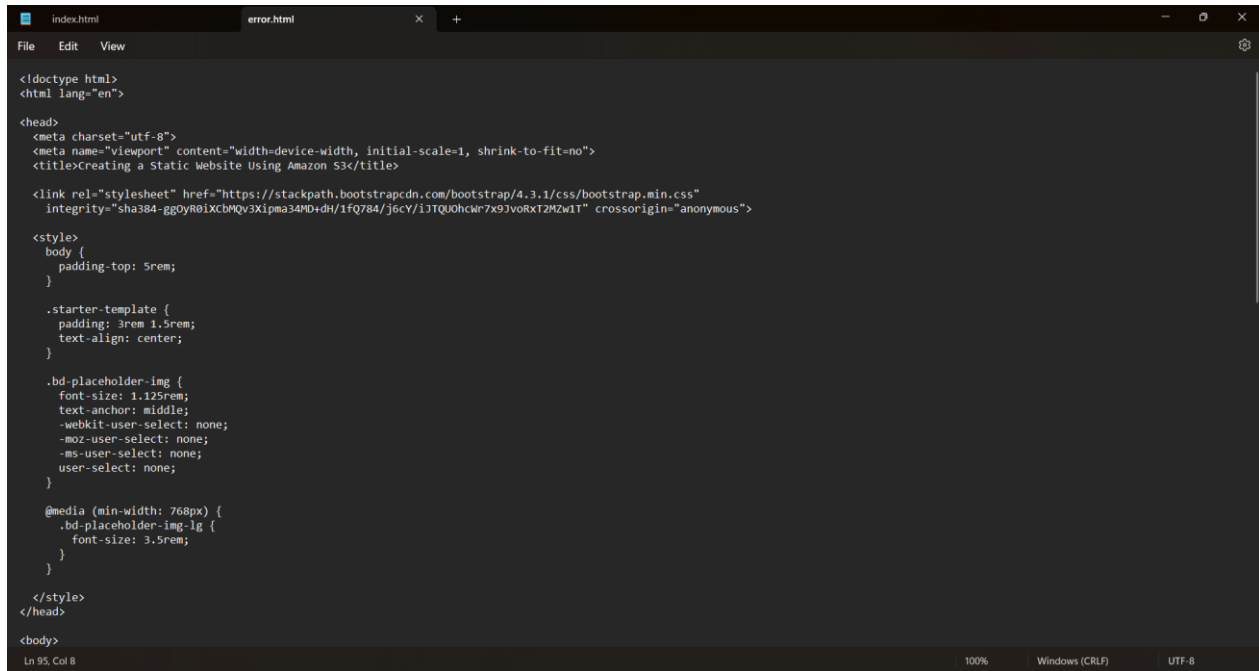
The screenshot shows the details of the 'my014567' bucket. The left sidebar is the same as the previous screenshot. The main content area displays the bucket's name, 'Publicly accessible' status, and tabs for Objects, Properties, Permissions, Metrics, Management, and Access Points. The 'Objects' tab is selected, showing a list of 2 objects:

Name	Type	Last modified	Size	Storage class
error.html	html	November 18, 2023, 08:59:23 (UTC+05:30)	3.6 KB	Standard
index.html/	Folder	-	-	-

The screenshot shows the Amazon S3 console interface. The left sidebar contains the navigation menu with options like Buckets, Access Points, Object Lambda Access Points, Multi-Region Access Points, Batch Operations, IAM Access Analyzer for S3, Storage Lens, Dashboards, Storage Lens groups, AWS Organizations settings, Feature spotlight, and AWS Marketplace for S3. The main content area displays the properties for bucket 'my014567'. It includes sections for Object Lock (Disabled), Requester pays (Disabled), and Static website hosting (Enabled). The static website hosting section shows the bucket website endpoint as <http://my014567.s3-website.ap-south-1.amazonaws.com>. The footer shows the AWS logo, CloudShell, Feedback, and copyright information for 2023.

The screenshot shows the Amazon S3 console interface, specifically the 'Permissions' tab for bucket 'my014567'. The left sidebar is the same as the previous screenshot. The main content area shows the 'Permissions overview' section, which includes a 'Public' access indicator. Below this, there is a 'Block public access (bucket settings)' section with an 'Edit' button. The 'Block all public access' setting is currently 'Off'. A link to 'Individual Block Public Access settings for this bucket' is provided. The footer shows the AWS logo, CloudShell, Feedback, and copyright information for 2023.





```
<!doctype html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
  <title>Creating a Static Website Using Amazon S3</title>

  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
        integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">

  <style>
    body {
      padding-top: 5rem;
    }

    .starter-template {
      padding: 3rem 1.5rem;
      text-align: center;
    }

    .bd-placeholder-img {
      font-size: 1.125rem;
      text-align: middle;
      -webkit-user-select: none;
      -moz-user-select: none;
      -ms-user-select: none;
      user-select: none;
    }

    @media (min-width: 768px) {
      .bd-placeholder-img-lg {
        font-size: 3.5rem;
      }
    }
  </style>
</head>

<body>
```

Ln 85, Col 8 100% Windows (CRLF) UTF-8

CONCLUSION

Deploying a static website using Amazon Simple Storage Service (S3) on the Amazon Web Services (AWS) platform represents a strategic choice that balances simplicity, cost-effectiveness, and scalability. Throughout this exploration, we have uncovered the motivations driving developers and businesses toward this deployment model and navigated the challenges inherent in this approach.

In conclusion, the deployment of a static website on AWS S3 offers a compelling solution for a diverse range of use cases. The simplicity of static websites, coupled with the reliability and scalability of AWS S3, empowers developers to create high-performance digital experiences. The cost-effectiveness of the pay-as-you-go model aligns with modern economic considerations, allowing organizations to optimize their infrastructure expenses.

While challenges such as limited dynamic functionality, workflow complexity, and SEO considerations exist, they can be effectively addressed through careful planning, configuration expertise, and a nuanced understanding of the specific needs of the project. The security features of AWS S3, combined with the integration of AWS IAM and SSL/TLS certificates, contribute to a robust security posture.

REFERENCES

<https://docs.aws.amazon.com/AmazonS3/latest/userguide/WebsiteHosting.html>

<https://docs.aws.amazon.com/AmazonS3/latest/userguide/HostingWebsiteOnS3Setup.html>