IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 3, March 2026

DOI: 10.17148/IJARCCE.2026.15364

Deploy a Static Website on AWS S3 and Use CloudFront

Yaswanth S[1], Mr. B. Ramesh Kumar[2]

III BCA, Department of Computer Applications, Sri Ramakrishna College of Arts & Science, Coimbatore, India[1]

Assistant Professor, Department of Computer Applications,

Sri Ramakrishna College of Arts & Science, Coimbatore, India[2]

Abstract: Cloud computing has transformed the way websites are hosted and delivered to users around the world. Traditional hosting methods often require complex infrastructure and maintenance. Amazon Web Services (AWS) provides scalable and cost-effective solutions to deploy websites quickly and efficiently.

This project demonstrates how to deploy a static website using Amazon S3 and improve its performance and security using Amazon CloudFront. Amazon S3 is used to store static website files such as HTML, CSS, and JavaScript, while CloudFront acts as a Content Delivery Network (CDN) that distributes the website content globally with low latency.

The implementation includes creating an S3 bucket, uploading website files, enabling static website hosting, configuring permissions, and integrating CloudFront for faster content delivery. The project highlights how cloud services simplify website deployment and enhance performance.

I. INTRODUCTION

In recent years, cloud computing has become a popular platform for hosting applications and websites. Organizations prefer cloud solutions because they provide high availability, scalability, and cost efficiency. Amazon Web Services (AWS) is one of the leading cloud platforms offering a variety of services for web hosting and deployment.

Static websites consist of fixed content such as HTML pages, CSS stylesheets, images, and JavaScript files. Unlike dynamic websites, static websites do not require server-side processing, making them simple and fast. AWS S3 provides an easy way to host static websites by storing files in a bucket and serving them directly to users.

Amazon CloudFront is a Content Delivery Network (CDN) that speeds up content delivery by caching website data at edge locations worldwide. By integrating CloudFront with S3, websites can load faster and provide a better user experience.
This project explains the process of deploying a static website on AWS S3 and using CloudFront to distribute the content globally.

II. RELATED WORK

Several researchers and organizations have explored cloud-based web hosting and content delivery techniques to improve website performance, scalability, and reliability. Cloud platforms such as Amazon Web Services (AWS) provide services like Amazon S3 and CloudFront that simplify the deployment and management of static websites.

A study on static website hosting using AWS explains that Amazon S3 can be used to store and serve static files such as HTML, CSS, JavaScript, and images. The research highlights that S3 offers a scalable and low-cost storage solution for hosting static websites while minimizing operational complexity.

Another research work on web content distribution using Amazon CloudFront discusses how Content Delivery Networks (CDNs) reduce latency by caching website content in multiple edge locations worldwide. This architecture ensures faster access to web resources and improves the user experience by delivering content from the nearest server location.

Researchers studying CDN architectures have also explained that proxy caching and distributed edge servers play a critical role in reducing bandwidth consumption and improving website performance. By storing frequently accessed content closer to users, CDNs significantly reduce response time and server load.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 3, March 2026
DOI: 10.17148/IJARCCE.2026.15364

Other studies on modern web infrastructure emphasize that CDNs are essential for large-scale web applications because they distribute content across geographically distributed servers, improving scalability and reliability. These networks ensure that users can access web content quickly regardless of their location.

AWS documentation further explains that static websites can be hosted directly on Amazon S3 by enabling static website hosting and serving content through a public endpoint. When integrated with Amazon CloudFront, the system benefits from caching, global distribution, and enhanced security features such as HTTPS support.

Based on these studies, deploying static websites using AWS S3 combined with CloudFront has become a widely accepted approach for building scalable, secure, and high-performance web applications.

III. SYSTEM ARCHITECTURE AND DESIGN

The system architecture for deploying a static website using Amazon Web Services consists of several cloud components that work together to deliver website content efficiently. The main components involved in this architecture are the user's web browser, Amazon CloudFront, and Amazon S3.

In this architecture, the static website files such as HTML, CSS, JavaScript, and images are stored inside an Amazon S3 bucket. Amazon S3 acts as the storage service and web hosting platform for the website. The static website hosting feature of S3 allows these files to be accessed through a public endpoint.

Amazon CloudFront is used as a Content Delivery Network (CDN) that sits between the users and the S3 bucket. CloudFront distributes the website content through multiple edge locations located around the world. When a user requests the website, the request is first sent to the nearest CloudFront edge server.

If the requested content is already cached in the CloudFront edge location, the content is delivered directly to the user, which reduces latency and improves loading speed. If the content is not cached, CloudFront retrieves it from the S3 bucket and then stores it temporarily in the edge location for future requests.

The user accesses the website through a web browser by entering the CloudFront distribution domain name. The browser sends the request to CloudFront, which then serves the website content efficiently using the global content delivery network.

This architecture improves website performance, scalability, and reliability. It also reduces the load on the origin server because frequently requested files are cached in multiple edge locations. As a result, users from different geographical locations can access the website quickly.

The overall system architecture ensures secure and high-performance website delivery while minimizing infrastructure management and operational costs.

IV. IMPLEMENTATION AND RESULTS

The implementation of this project focuses on deploying a static website using Amazon S3 and improving its performance with Amazon CloudFront. The process involves creating an S3 bucket, uploading the website files, enabling static website hosting, and configuring a CloudFront distribution to deliver the content globally.

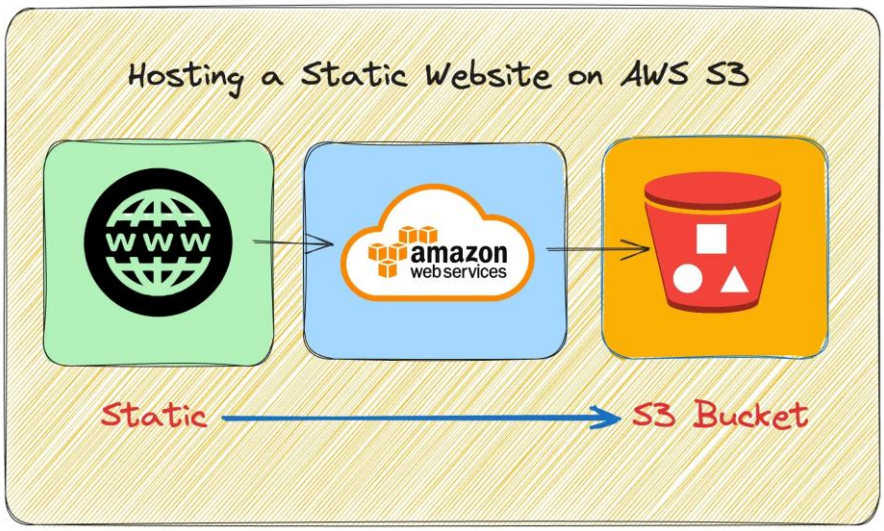
The first step in the implementation process is creating an AWS account and accessing the AWS Management Console. After logging in, the Amazon S3 service is selected to create a new storage bucket. The bucket name must be unique and is used to store the website files such as HTML, CSS, JavaScript, and images.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 3, March 2026

DOI: 10.17148/IJARCCE.2026.15364



After creating the bucket, the **Block Public Access** settings are modified to allow the website files to be publicly accessible. Then the **Static Website Hosting** option is enabled in the bucket properties. In this configuration, the main page of the website is defined as the **index.html** document.

Next, the website files are uploaded to the S3 bucket using the upload feature available in the console. Once the files are uploaded, a **bucket policy** is added to allow public read access so that users can access the website through the internet. After enabling static website hosting, AWS generates a **website endpoint URL**. This URL can be used to access the deployed static website directly from the S3 bucket.

To improve performance and global availability, Amazon CloudFront is configured. A new CloudFront distribution is created, and the S3 bucket is selected as the origin source. CloudFront caches the website content in multiple edge locations around the world.

Once the distribution is deployed, CloudFront generates a **distribution domain name** that acts as the public access URL for the website. Users accessing this URL receive content from the nearest edge location, which significantly reduces latency and improves loading speed.

The results of the implementation show that the static website was successfully deployed on AWS infrastructure. The website could be accessed through both the S3 endpoint and the CloudFront distribution domain. When accessed through CloudFront, the website loading time was faster due to content caching and optimized delivery.

The use of Amazon CloudFront improved website performance by reducing response time and distributing the content globally. This implementation demonstrates that cloud-based hosting using AWS services provides a reliable, scalable, and efficient method for deploying static websites.

V. SYSTEM MODULES

The proposed system is divided into several modules to simplify the process of deploying and managing a static website using cloud services. Each module performs a specific function to ensure efficient deployment and delivery of website content.

1. User Access Module

This module represents the interaction between the user and the deployed website. Users access the website through a web browser by entering the website URL or the CloudFront distribution domain. The browser sends a request to the content delivery network to retrieve the website files.

2. Amazon S3 Bucket Creation Module

This module is responsible for creating and configuring an Amazon S3 bucket that stores the static website files. The bucket acts as the primary storage location where HTML, CSS, JavaScript, and image files are uploaded and managed.

3. Static Website Hosting Module

In this module, the static website hosting feature of Amazon S3 is enabled. This configuration allows the files stored in

DOI: 10.17148/IJARCCE.2026.15364

the bucket to be served as a website through a public endpoint. The index document and error document are also specified in this module.

4. File Upload and Management Module

This module handles the uploading and organization of website files in the S3 bucket. The project files such as HTML pages, style sheets, and images are uploaded using the AWS Management Console. Proper file management ensures that the website functions correctly when accessed by users.

5. Security and Permission Module

This module manages access permissions and security settings. Bucket policies and public access settings are configured so that the website files can be accessed by users through the internet while maintaining appropriate security controls.

6. CloudFront Distribution Module

This module integrates Amazon CloudFront with the S3 bucket. CloudFront acts as a Content Delivery Network (CDN) that caches website content at edge locations around the world. It ensures faster delivery of website content to users regardless of their geographical location.

7. Content Delivery Module

This module handles the delivery of cached content from CloudFront edge servers to the user. If the requested content is already cached, it is delivered immediately. Otherwise, CloudFront retrieves it from the S3 bucket and then serves it to the user.

VI. EXPERIMENTAL RESULTS AND PERFORMANCE ANALYSIS

The experimental evaluation of the proposed system was conducted after successfully deploying the static website using Amazon S3 and integrating it with Amazon CloudFront. The main objective of the experiment was to observe the performance of the hosted website and analyze the improvement achieved through the use of cloud-based content delivery services.

After completing the deployment process, the website files were stored in the Amazon S3 bucket and static website hosting was enabled. The website was initially accessed using the S3 website endpoint to verify that the files were correctly uploaded and publicly accessible. The results showed that the website loaded successfully and all the static elements such as HTML pages, CSS styles, and images were displayed properly.

In the next stage, Amazon CloudFront was configured to distribute the website content. The S3 bucket was set as the origin for the CloudFront distribution, allowing CloudFront to retrieve and cache website files. Once the distribution was deployed, the website was accessed using the CloudFront domain name.

The performance analysis showed a noticeable improvement in website loading time when accessed through CloudFront. This improvement occurs because CloudFront caches the website content in multiple edge locations around the world. When a user requests the website, the request is routed to the nearest edge server, which reduces the time required to retrieve the content.

Another important result observed during the experiment was the reduction in server load on the origin S3 bucket. Since frequently accessed files were cached in the CloudFront edge servers, repeated requests were served directly from the cache rather than fetching data from the origin server each time. This caching mechanism improves efficiency and reduces bandwidth usage.

The system also demonstrated high availability and scalability. Even if multiple users accessed the website simultaneously, the content delivery network ensured smooth performance by distributing the traffic across multiple servers. This capability makes the system suitable for websites that expect high traffic.

VII. CONCLUSION

This project demonstrated the process of deploying a static website using Amazon S3 and improving its performance through Amazon CloudFront. The implementation showed how cloud computing services can simplify the process of hosting and delivering web content without the need for traditional web servers or complex infrastructure.

By using Amazon S3, the static website files such as HTML, CSS, and JavaScript were stored and hosted efficiently. The configuration of static website hosting allowed users to access the website directly through a public endpoint. The integration of Amazon CloudFront further enhanced the system by distributing the website content through a global network of edge locations.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 3, March 2026
DOI: 10.17148/IJARCCE.2026.15364

ACKNOWLEDGMENT

I would also like to thank my institution for providing the necessary resources and opportunities to work on this project. The facilities and learning environment provided by the college played an important role in the successful completion of this work.

I extend my heartfelt thanks to my friends and classmates for their support and cooperation during the development of this project. Their discussions and suggestions helped in improving the quality of the project.

REFERENCES

[1]. 1.Amazon Web Services, *Amazon S3 Documentation*. Available at: https://aws.amazon.com/s3/
[2]. 2.Amazon Web Services, *Amazon CloudFront Developer Guide*. Available at: https://aws.amazon.com/cloudfront/
[3]. 3.Amazon Web Services Documentation, *Hosting a Static Website on Amazon S3*. Available at: https://docs.aws.amazon.com/AmazonS3/latest/userguide/WebsiteHosting.html
[4]. 4.Rajkumar Buyya, Christian Vecchiola, and Thamarai Selvi, *Mastering Cloud Computing: Foundations and Applications Programming*. McGraw-Hill Education.
[5]. 5.Thomas Erl, Ricardo Puttini, and Zaigham Mahmood, *Cloud Computing: Concepts, Technology and Architecture*. Pearson Education.
[6]. 6.Research papers and articles related to Content Delivery Networks (CDN) and cloud-based web hosting from IEEE and other academic sources.
[7]. 7.Online tutorials and technical resources related to AWS S3 and CloudFront implementation.