

A Course End Project (Cloud Computing)  
on

## **Static Web Hosting in AWS**

Submitted in the Partial Fulfillment of the  
Requirements  
for the Award of the Degree of

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE AND ENGINEERING (AI&ML)**

Submitted

By

**Team No.: 04**

**Anupama Amarender**

**21881A6670**

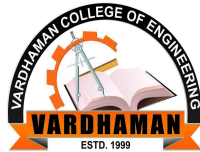
**HariPriya Koratala**

**21881A6697**

Under the Esteemed Guidance of

**Dr. P. Pavan Kumar**

**Associate Professor**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (AI&ML)**

**VARDHAMAN COLLEGE OF ENGINEERING**

**(AUTONOMOUS)**

Affiliated to **JNTUH**, Approved by **AICTE**, Accredited by **NAAC**, with **A++** Grade, **ISO 9001:2015** Certified  
Kacharam, Shamshabad, Hyderabad – 501218, Telangana, India

**2023- 24**

## ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of the task would be put incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crown all the efforts with success.

We wish to express my deep sense of gratitude to **Dr. P. Pavan Kumar, Associate Professor** for their able guidance and useful suggestions, which helped us in completing the design part of potential project in time.

We are particularly thankful to **Dr. M. A. Jabbar**, Professor & Head, Department of Computer Science and Engineering (AI&ML) for his guidance, intense support and encouragement, which helped us to mould our project into a successful one.

We show gratitude to our honorable Principal **Dr. J. V. R. Ravindra**, for having provided all the facilities and support.

We avail this opportunity to express our deep sense of gratitude and heartfelt thanks to **Dr. Teegala Vijender Reddy**, Chairman and **Sri Teegala Upender Reddy**, Secretary of VCE, for providing congenial atmosphere to complete this project successfully.

We also thank all the staff members of the department of **CSE(AI&ML)** for their valuable support and generous advice. Finally, thanks to all our friends and family members for their continuous support and enthusiastic help.

**Anupama Amarender - 21881A6670**

**Haripriya Koratala - 21881A6697**



# **VARDHAMAN COLLEGE OF ENGINEERING**

**(AUTONOMOUS)**

Affiliated to JNTUH, Approved by AICTE, Accredited by NAAC, with A++ Grade, ISO 9001:2015 Certified  
Kacharam, Shamshabad, Hyderabad – 501218, Telangana, India

## **Department of Computer Science and Engineering (AI&ML)**

### **CERTIFICATE**

This is to certify that the Course End Project report work entitled “**Static Web Hosting in AWS**” carried out by Ms. Anupama Amarender, Roll Number **21881A6670**, Ms. Haripriya Koratala, Roll Number **21881A6697** towards Course End Project and submitted to the Department of Computer Science and Engineering(AI&ML), in partial fulfillment of the requirements for the award of degree of **Bachelor of Technology in Computer Science and Engineering (AI&ML)** during the year 2023-24.

**Name & Signature of the Instructors**

**Dr. P. Pavan Kumar**

**Associate Professor**

**Name & Signature of the HOD**

**Dr M A Jabbar**

**HOD, CSE(AI&ML)**

## **ABSTRACT**

This project explores the implementation of static web hosting using Amazon Web Services (AWS) Simple Storage Service (S3). By utilizing AWS S3, we can host static websites, which are ideal for sites that do not require server-side processing. The focus is on creating and deploying an 'index.html' file for the main content and an 'error.html' file to handle error messages. This method ensures a cost-effective, scalable, and reliable solution for web hosting. The project demonstrates the capability of AWS S3 to serve static websites to a global audience with minimal setup and maintenance, leveraging AWS's robust infrastructure to ensure high availability and performance.

The project demonstrates how AWS S3 can be configured to serve static websites to a global audience with minimal setup and maintenance. By enabling static web hosting on an S3 bucket and setting appropriate bucket policies, we can provide public access to the website, making it accessible via a simple URL. This method provides a straightforward and efficient way to deploy and manage web content, ensuring users can take advantage of AWS's scalability and reliability for their static web hosting needs.

## ABBREVIATIONS

Abbreviation	Expansion
AWS	Amazon Web Services
S3	Simple Storage Service
CSS	Cascading Style Sheets
HTML	Hypertext Markup Language
CLI	Command Line Interface
HTTPS	Hypertext Transfer Protocol Secure
CDN	Content Delivery Network
SSL/TLS	Secure Sockets Layer / Transport Layer Security
CI/CD	Continuous Integration / Continuous Deployment
CD	Continuous Deployment
IAM	Identity and Access Management
DDoS	Distributed Denial of Service
CI	Continuous Integration

## Table of Contents

<b>Chapter No.</b>	<b>Title</b>	<b>Page No.</b>
	Acknowledgements	2
	Abstract	4
	Abbreviations	5
<b>Chapter 1</b>	<b>Introduction</b>	7
	1.1 Scope	7
	1.2 Objectives	7
<b>Chapter 2</b>	<b>Problem Definition and Proposed System Methodology</b>	8
	2.1 Problem Statement	8
	2.2 Proposed System Methodology	8
	2.3 Code	10
<b>Chapter 3</b>	<b>Software Requirements Specification and Hardware Requirements</b>	12
<b>Chapter 4</b>	<b>System Design Diagrams</b>	14
<b>Chapter 5</b>	<b>Results and Discussions</b>	15
<b>Chapter 6</b>	<b>Conclusion and Future Scope</b>	16
	<b>References</b>	18

# CHAPTER-1

## Introduction

### 1.1 Scope

Static web hosting encompasses the deployment of websites that consist of fixed content, typically HTML, CSS, JavaScript, and media files. The scope of this project includes leveraging Amazon Web Services (AWS) for hosting static websites, focusing on AWS S3 for scalable storage. The project aims to explore the efficiency, scalability, and cost-effectiveness of using AWS services for static web hosting, catering to developers and businesses seeking reliable web hosting solutions..

### 1.2 Objectives

The objectives of this project are:

- Create HTML files: 'index.html' and 'error.html'
- Set up AWS S3 bucket.
- Configure public access.
- Enable static web hosting.
- Upload HTML files to S3 bucket.
- Test the static website.
- Document the process.

## CHAPTER-2

### Problem Definition and Proposed System Methodology

#### 2.1 Problem Statement

The objective of this project is to create a static website hosted on AWS S3. This involves developing two HTML files(index.html and error.html), configuring an S3 bucket to serve these files, and ensuring that the website is publicly accessible. Additionally, the project aims to demonstrate the steps required to set up static web hosting on AWS S3, configure appropriate access policies, and validate the functionality of the hosted website. The final outcome will be a fully functional static website accessible via a provided URL, with error handling for invalid paths.

#### 2.2 Proposed System Methodology

##### 1.Create HTML Files:

index.html: This file will contain a hyperlink.

error.html: This file will display an error message.

##### 2.Log in to AWS Management Console:

Open the AWS Management Console and log in to your account.

##### 3.Create an S3 Bucket:

Navigate to the S3 service under the Storage section.

Click on "Create bucket" and provide an appropriate name for the bucket (e.g., mybucket109).

Uncheck the "Block all public access" option to allow public access to the static website.

Click on "Create bucket."

##### 4.Configure Static Web Hosting:

Go to the created bucket and click on the "Properties" tab.

Under "Static web hosting," enable static web hosting.



Specify the index document as index.html and the error document as error.html.

Click on "Save changes."

#### 5.Set Bucket Policy:

Navigate to the "Permissions" tab and click on "Edit" under "Bucket policy."

Copy and paste the following policy and click on "Save changes":

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowPublicReadAccess",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::mybucket109/*"
    }
  ]
}
```

#### 6.Upload HTML Files:

Go to the "Objects" tab and click on "Upload."

Add the files index.html and error.html, then click on "Upload."

#### 7.Access the Website:

Go to the "Properties" tab, then "Static web hosting."

Copy the provided URL and paste it into a new browser tab.

The index.html page should be displayed. If any additional text is added to the URL, the error message from error.html will be displayed.

## 2.2 Code

index.html

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
  <title>Basic HTML Page</title>
```

```
</head>
```

```
<body>
```

```
  <h1>Welcome to My Basic HTML Page</h1>
```

```
  <p>Click the link below to go to Google:</p>
```

```
  <a href="https://www.google.com" target="_blank">Go to Google</a>
```

```
</body>
```

```
</html>
```

error.html

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
  <title>Error Page</title>
```

```
</head>
```

```
<body>
```

```
  <h1>Error</h1>
```

</body>

</html>

## CHAPTER-3

### Software Requirements Specification and Hardware Requirements

Software Requirements:

AWS Account:

- An active AWS account is required to access AWS services, including S3.

AWS Management Console:

- Access to the AWS Management Console through a web browser to configure and manage AWS services.

Text Editor:

- A text editor (e.g., VS Code, Sublime Text) to create and edit HTML files (index.html and error.html)

Web Browser:

- A modern web browser (e.g., Chrome, Firefox, Safari) to test and view the hosted static website.

AWS CLI (Optional):

- The AWS Command Line Interface can be used for advanced configurations and automation tasks, but it's optional for basic setup.

HTML, CSS, and JavaScript Knowledge:

- Basic understanding of HTML for creating web pages.
- Optional knowledge of CSS and JavaScript for enhancing the website's appearance and functionality.

Internet Connection:

- Stable internet connection to access AWS services and upload files to S3.

## Hardware Requirements

### Personal Computer or Device:

- A computer or device with an internet connection to access the AWS Management Console and upload files to AWS S3.

### Operating System:

- Compatible with major operating systems such as Windows, macOS, or Linux.

### Web Browser:

- A modern web browser (e.g., Chrome, Firefox, Safari) to access the AWS Management Console and test the hosted static website.

### Internet Connection:

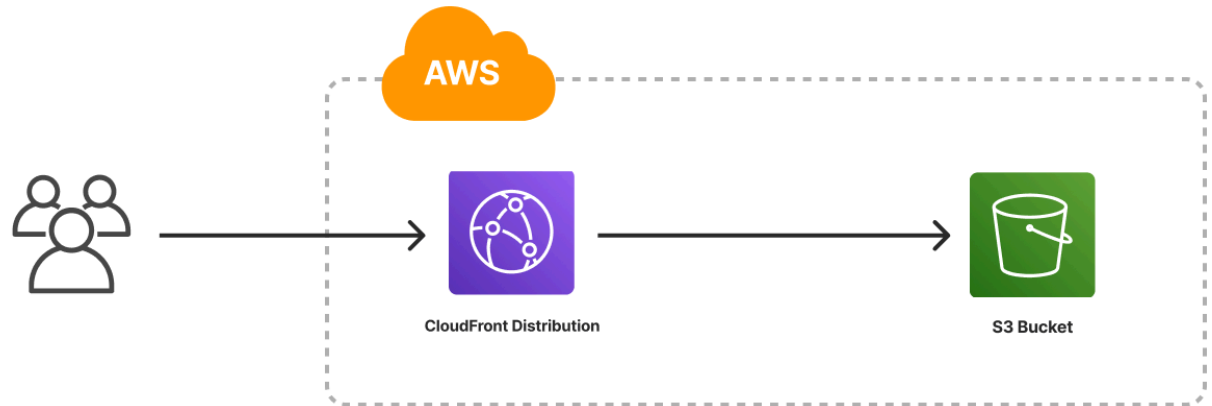
- Stable and reliable internet connection to access AWS services and upload files to AWS S3.

### Optional:

- For advanced configurations and automation tasks, a computer with sufficient processing power and memory to handle AWS CLI operations efficiently.

## CHAPTER-4

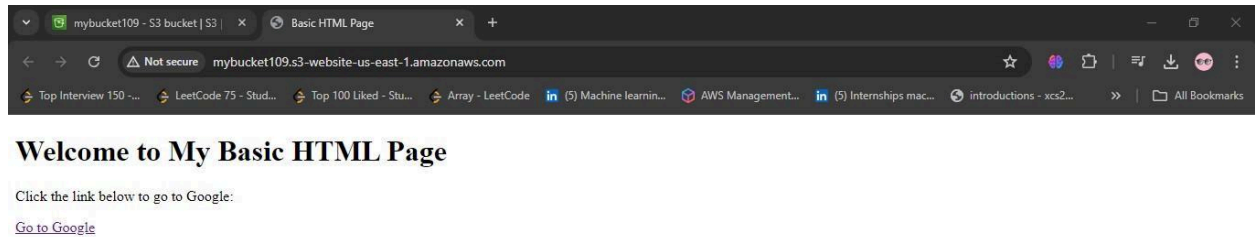
### System Design Diagrams



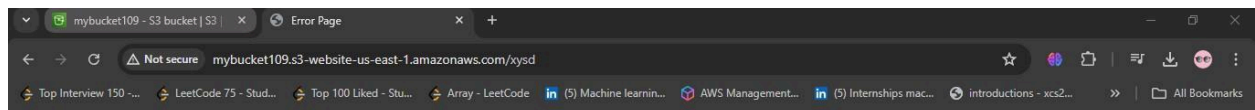
## CHAPTER-5

### Results and Discussions

On copying and pasting the provided URL the index.html page is displayed as follows



But by changing the url to include additional characters the error.html page is displayed with the error message as follows



**Error**

## CHAPTER-6

### Conclusion and Future Scope

Hosting a static website on AWS S3 provides a straightforward and cost-effective solution for making web content accessible to users worldwide. This project has demonstrated the essential steps involved in setting up such a website, from creating HTML files to configuring AWS services. By following these steps, users can ensure their static websites are secure, scalable, and easily maintainable.

Future scope:

Enhanced Website Features:

- Explore integrating additional features using JavaScript frameworks like React or Angular to enhance user interaction and dynamic content.

Custom Domain and SSL/TLS Integration:

- Configure custom domain names and enable SSL/TLS certificates to secure connections and enhance trustworthiness.

Content Delivery Network (CDN) Integration:

- Utilize AWS CloudFront or other CDN services to improve website performance by caching content closer to users globally.

Monitoring and Analytics:

- Implement AWS CloudWatch for monitoring website metrics and logs, providing insights into performance and user behavior.

Automated Deployment:

- Implement CI/CD pipelines using AWS CodePipeline and AWS CodeBuild for automated testing, building, and deploying updates to the website.

Scaling Options:



- Explore AWS Auto Scaling capabilities to automatically adjust resources based on traffic fluctuations, ensuring optimal performance during peak times.

Security Enhancements:

- Continuously review and enhance security measures, such as AWS WAF for web application firewall protection and AWS Shield for DDoS mitigation.

These future scope points aim to enhance the project's capabilities, scalability, security, and operational efficiency, leveraging AWS's comprehensive cloud services for dynamic and evolving requirements in static web hosting.

## References

1. <https://docs.aws.amazon.com/AmazonS3/latest/userguide/WebsiteHosting.html>
2. <https://www.geeksforgeeks.org/how-to-host-static-website-using-aws-s3/>
3. <https://medium.com/@kyle.galbraith/how-to-host-a-website-on-s3-without-getting-lost-in-the-sea-e2b82aa6cd38>
4. <https://medium.com/@sudheer.barakers/step-by-step-guide-hosting-a-static-website-on-a-ws-s3-660b81bea177>