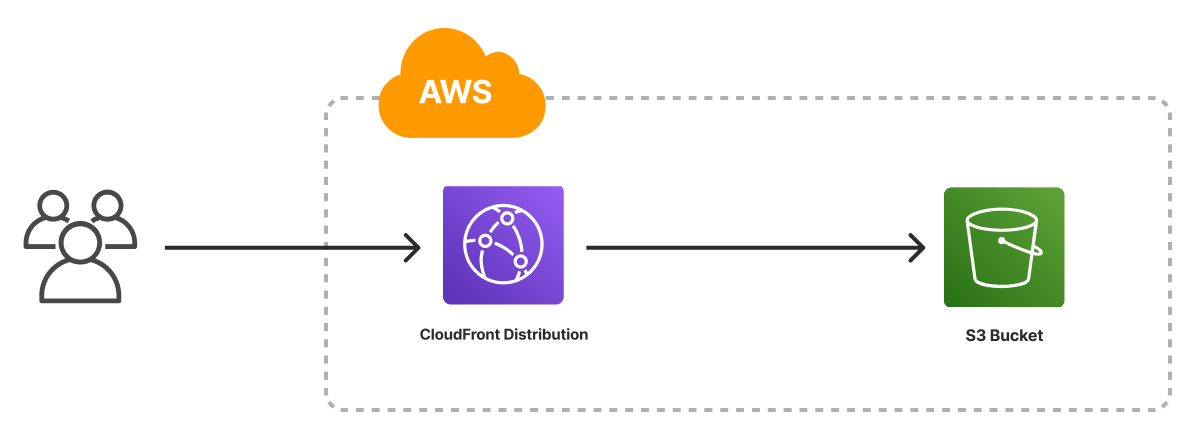
**Project 2   
Deploy a Static Website on AWS**



Name- Tanuja Tiwari

Table of Contents

* **Introduction**
* **Objective**
* **Abstract of the Project**
* **Services used**
* **Implementation**
* **Task 1: Sign in to AWS Management Console**
* **Task 2: Creating a S3 Bucket**
* **Task 3: Enable Static Website Hosting settings**
* **Task 4: Test the website**
* **Task 5: Test the website's error page**

**Conclusion**



* **Introduction**

**In today's digital world, having a fast, scalable, and cost-effective web presence is essential. Static websites—sites made up of fixed content like HTML, CSS, JavaScript, and media files—are ideal for portfolios, documentation, marketing pages, and more. Amazon Web Services (AWS) provides a powerful, reliable, and highly available infrastructure to host static websites with ease.**

* **Objective**

**The objective of this project is to deploy a fully functional static website using Amazon Web Services (AWS). This involves hosting the website’s files—such as HTML, CSS, JavaScript, and images—on Amazon S3, configuring the necessary permissions for public access, and optionally enhancing performance and availability using AWS CloudFront**

* **Abstract of this project**

**This project focuses on deploying a static website using Amazon Web Services (AWS), leveraging its scalable and reliable cloud infrastructure. The website consists of static assets such as HTML, CSS, JavaScript, and images, which are hosted using Amazon S3 (Simple Storage Service). The setup involves configuring S3 buckets for web hosting, managing permissions for public access, and optionally integrating AWS CloudFront for faster global content delivery and Route S3 for custom domain management. The project demonstrates a practical, cost-effective solution for hosting static websites with high availability, performance, and minimal maintenance, making it ideal for personal portfolios, business landing pages, or documentation sites**.

* **Services used**
* **S3**
* **Implementation**

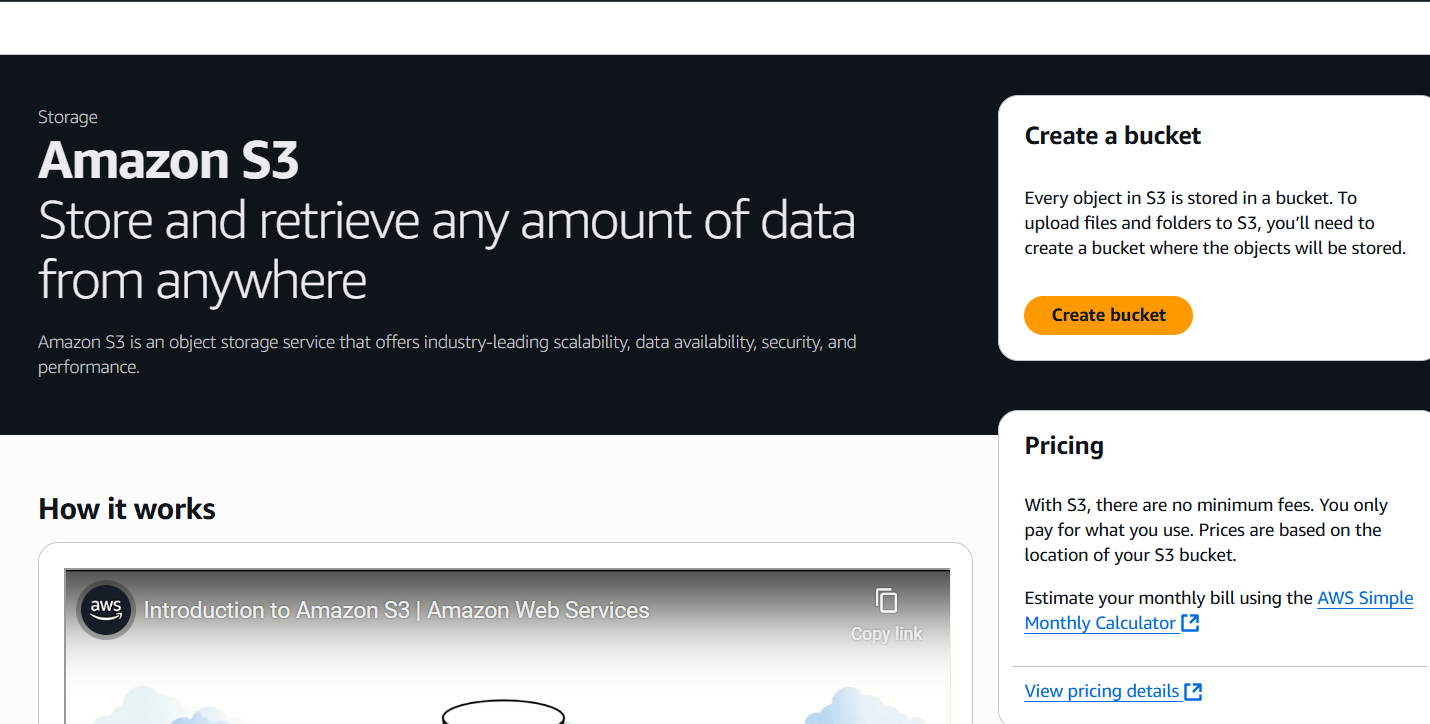
**Task 1: Sign in to AWS Management Console**

1. **Click on the Sign in to console, and you will get redirected to AWS Console in a new browser tab.**
2. **On the AWS sign-in page,**
   * **Leave the Account ID as default. Never edit/remove the 12-digit Account ID present in the AWS Console. otherwise, you cannot proceed with the lab.**
   * **Now enter your username and password in AWS Console and click on the Sign in button.**
3. **Once Signed in to the AWS Management Console, Make the default AWS Region as ap south-1.**

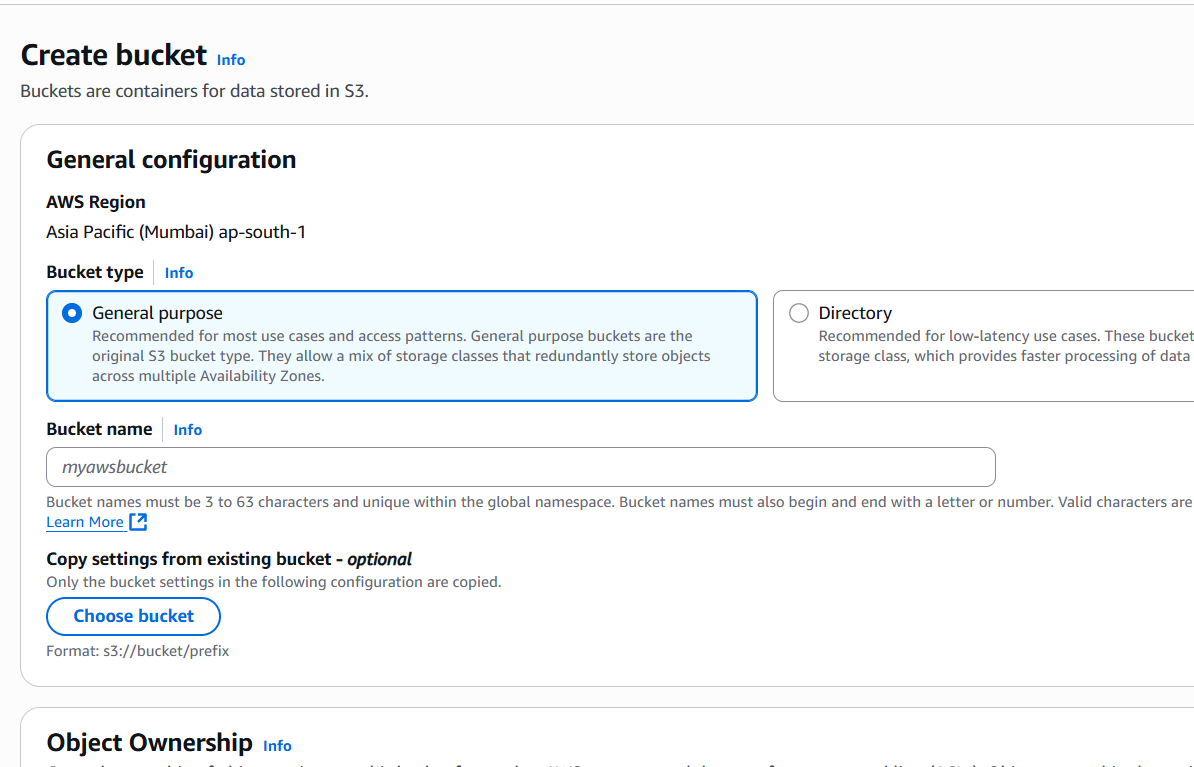
**Task 2: Creating a S3 Bucket**

**In this task, we are going to create a new S3 bucket in the US East (N. Virginia) region with a unique name disabling ACLs, and allowing public access for hosting the static website.**

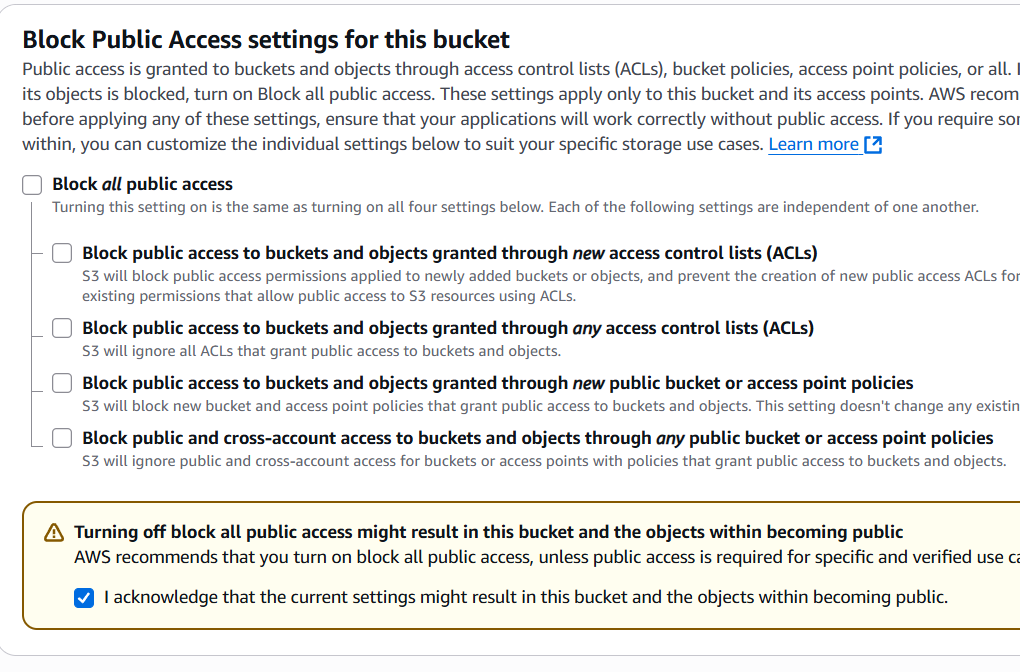
1. **Navigate to S3 by clicking on the Services menu at the top, then click on S3 in the Storage section.**
2. **In the S3 dashboard, click on the Create Bucket button.**

****

1. **In the General Configuration, bucket name: Enter abcxyz**
   * **Note: Enter a unique name because bucket names are globally unique.**
2. **AWS Region: ap-south-01**

****

1. **Object ownership: Select ACLs disabled (recommended) option**
2. **In the option of Block Public Access settings for this bucket, Uncheck the option of Block all public access.** 
   * **Check the I acknowledge that the current settings might result in this bucket and the objects within becoming public checkbox.**

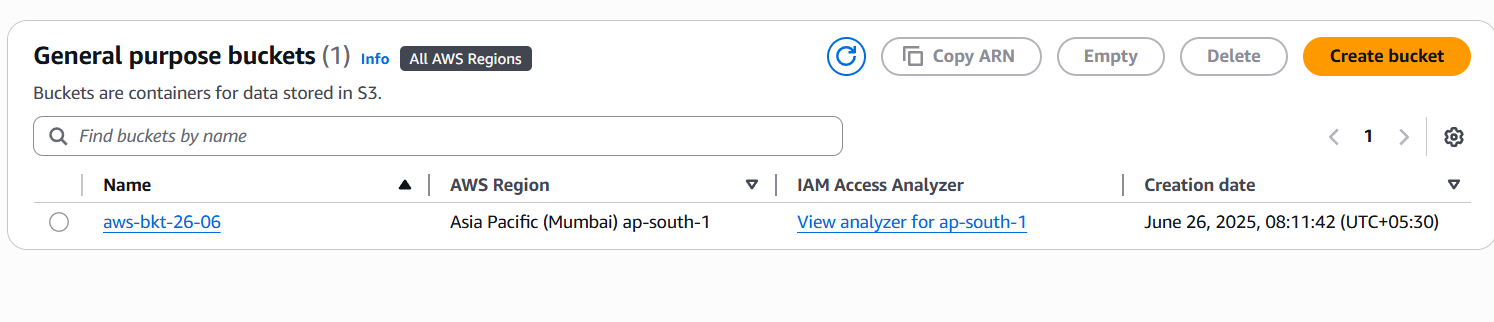
****

1. **Keep everything default and click on Create Bucket button.**

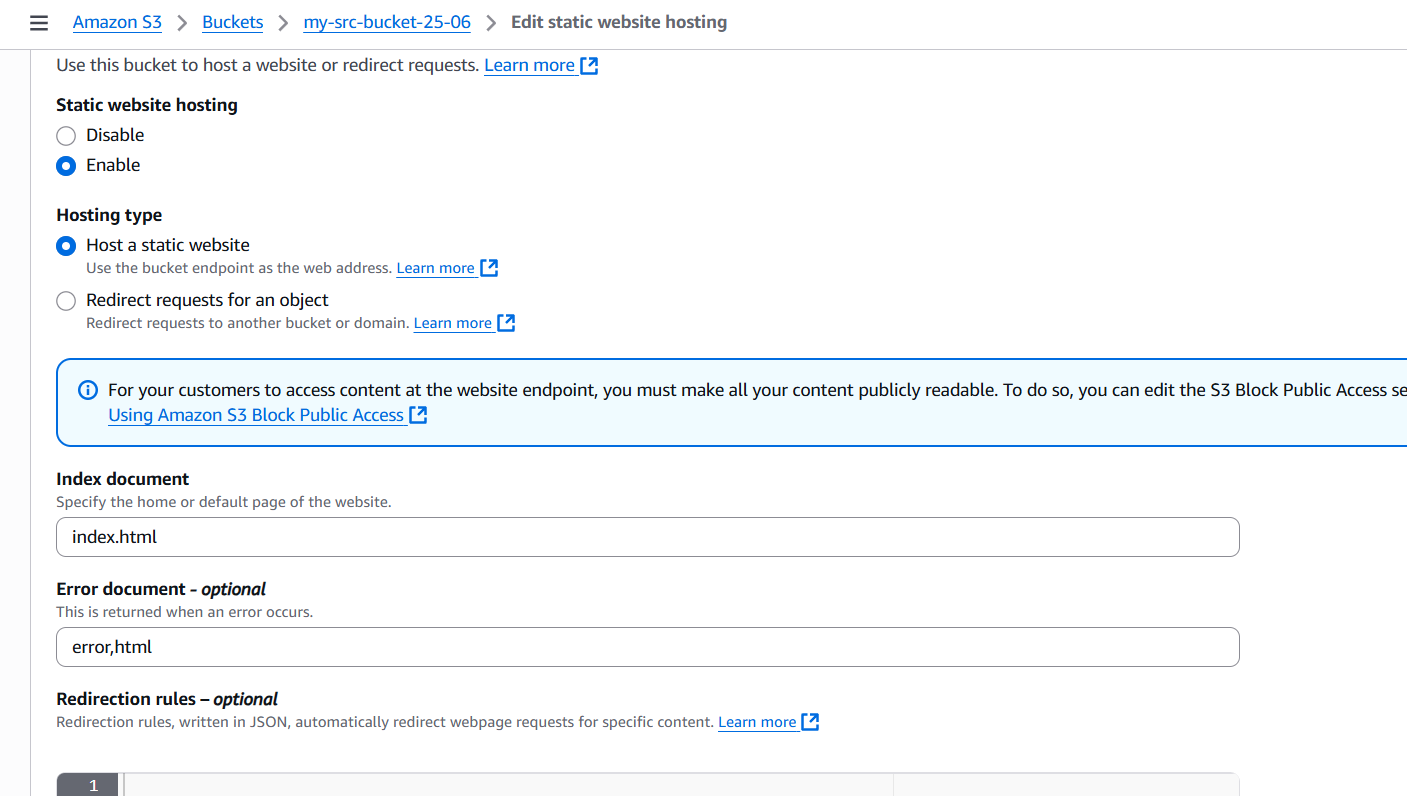
**Task 3: Enable Static Website Hosting settings**

**In this task, we will enable static website hosting for our S3 bucket, configure it to use index.html and error.html, copy the provided endpoint, upload two files, and configure the bucket policy by copying its ARN and pasting the provided policy code.**

1. **To proceed, go to the S3 bucket name that you created and click on it. After that, navigate to the Properties tab which can be found at the top of the screen.**

****

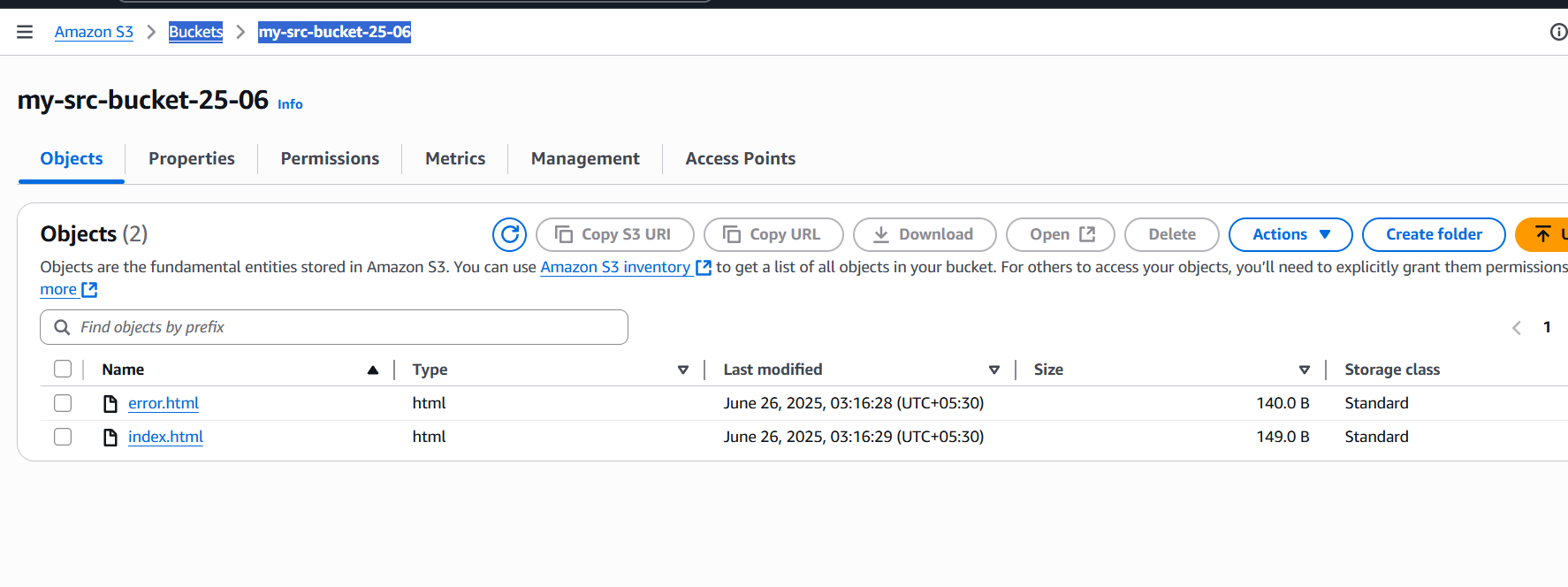
1. **Scroll down to the Static website hosting section and click on Edit button.**

****

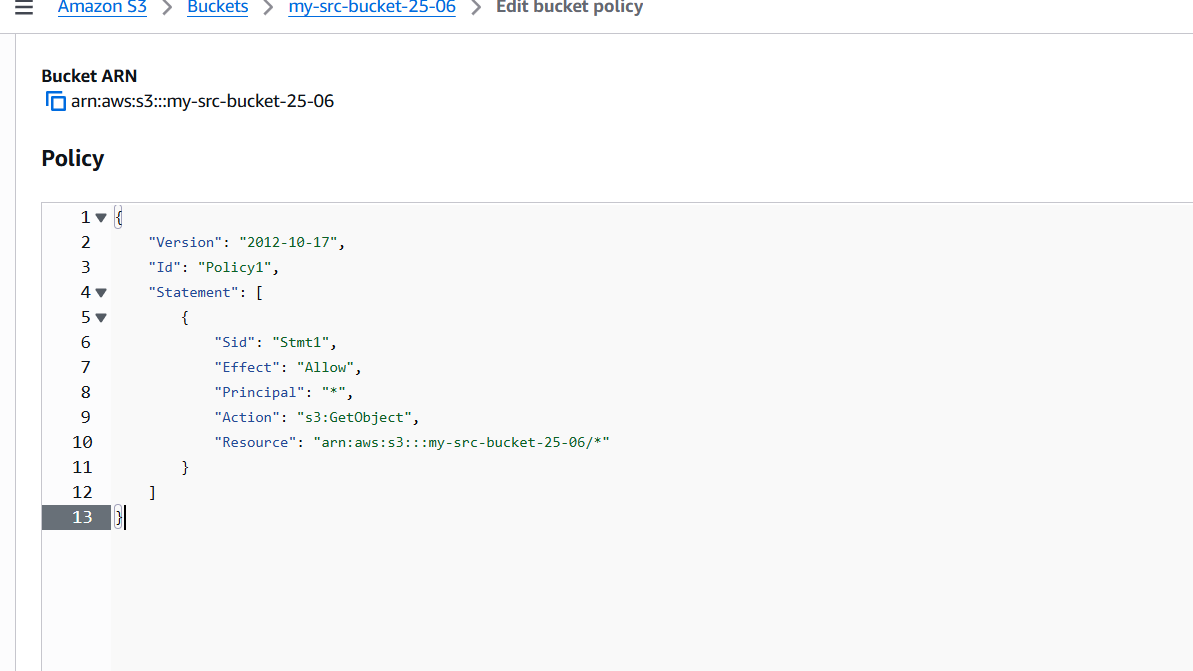
1. **In the Static website hosting dialog box**

* **Static website hosting: Select Enable**
* **Hosting type: Choose Host a static website**
* **Index document: Type *index.html***
* **Error document: Type *error.html***
* **Click on Save Changes.**

1. **Go to the Properties tab of your S3 bucket, and find the Static website hosting section. Copy the Endpoint provided in this section to your clipboard and save it for future reference.**
2. **The next step is to download the zip file by clicking on the** [**link**](https://labresources.whizlabs.com/2e8a0c891334fbb141d713fc9de61e60/task_23.zip)**, extract it, and upload two files named index.html and error.html to the S3 bucket you created earlier.**

****

1. **To configure your S3 bucket, access the Permissions tab and make the necessary configurations.**

**In the Permissions tab, Click on the Edit button beside the Bucket Policy.  
**

* **In the policy above, Update the bucket ARN on the Resource key value and paste the above policy code in the editor.**

1. **Click on Save changes button**.

**Task 4: Test the website**

**Now copy the static website URL (that we saved earlier) and run it in your browser. You will be able to see the index.html file's text. A sample screenshot is attached below:**

**Task 5: Test the website's error page**

**Copy the static website URL (which we saved earlier), but this time, add some random characters to the end of the URL to break it. When satisfied, hit [Enter]. You will be redirected to the error.html page automatically.**

* **Conclusion**

**In conclusion, deploying a static website on AWS provides an efficient, secure, and scalable solution for hosting modern web content. By utilizing Amazon S3 for storage, CloudFront for content delivery, and Route 53 for domain management, this project demonstrates how cloud services can simplify the process of launching and maintaining a website. The experience gained through this deployment not only highlights the power and flexibility of AWS but also equips users with valuable cloud computing skills applicable to real-world scenarios. Overall, this approach offers a cost-effective alternative to traditional web hosting, making it ideal for developers, students, and businesses alike.**