**Create a Static Website with Amazon Simple Storage Service (Amazon S3)**

**Lab Topics**

* [Lab overview](https://labs.skillbuilder.aws/sa/lab/arn%3Aaws%3Alearningcontent%3Aus-east-1%3A470679935125%3Ablueprintversion%2FSPL-BE-100-DVCSWS-1%3A1.0.2-08cfa5b1/en-US#lab-overview)
* [Start lab](https://labs.skillbuilder.aws/sa/lab/arn%3Aaws%3Alearningcontent%3Aus-east-1%3A470679935125%3Ablueprintversion%2FSPL-BE-100-DVCSWS-1%3A1.0.2-08cfa5b1/en-US#start-lab)
* [Task 1: Create an S3 bucket](https://labs.skillbuilder.aws/sa/lab/arn%3Aaws%3Alearningcontent%3Aus-east-1%3A470679935125%3Ablueprintversion%2FSPL-BE-100-DVCSWS-1%3A1.0.2-08cfa5b1/en-US#task-1-create-an-s3-bucket)
* [Task 2: Configure the S3 bucket as a static website and allow public access](https://labs.skillbuilder.aws/sa/lab/arn%3Aaws%3Alearningcontent%3Aus-east-1%3A470679935125%3Ablueprintversion%2FSPL-BE-100-DVCSWS-1%3A1.0.2-08cfa5b1/en-US#task-2-configure-the-s3-bucket-as-a-static-website-and-allow-public-access)
* [Task 3: Add a bucket policy to allow public access to content in our bucket](https://labs.skillbuilder.aws/sa/lab/arn%3Aaws%3Alearningcontent%3Aus-east-1%3A470679935125%3Ablueprintversion%2FSPL-BE-100-DVCSWS-1%3A1.0.2-08cfa5b1/en-US#task-3-add-a-bucket-policy-to-allow-public-access-to-content-in-your-bucket)
* [Task 4: Create and upload the website assets and test the website](https://labs.skillbuilder.aws/sa/lab/arn%3Aaws%3Alearningcontent%3Aus-east-1%3A470679935125%3Ablueprintversion%2FSPL-BE-100-DVCSWS-1%3A1.0.2-08cfa5b1/en-US#task-4-create-and-upload-the-website-assets-and-test-the-website)
* [End lab](https://labs.skillbuilder.aws/sa/lab/arn%3Aaws%3Alearningcontent%3Aus-east-1%3A470679935125%3Ablueprintversion%2FSPL-BE-100-DVCSWS-1%3A1.0.2-08cfa5b1/en-US#end-lab)

**Lab overview**

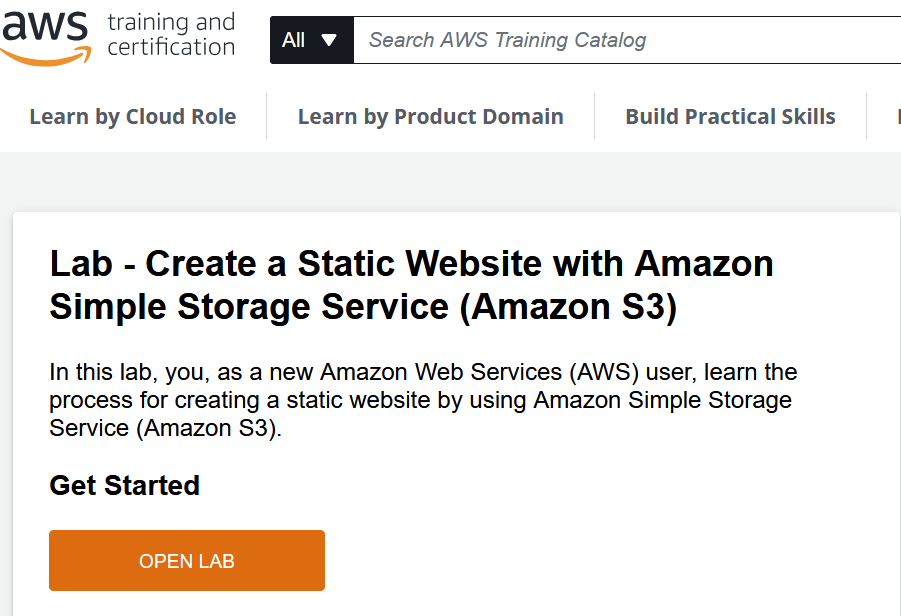
In this lab, we as a new Amazon Web Services (AWS) user, learn the process for creating a static website by using Amazon Simple Storage Service (Amazon S3). We learn how to configure an S3 bucket policy, which is required to allow public access.

****

**Objectives**

By this lab, we are able to do the following:

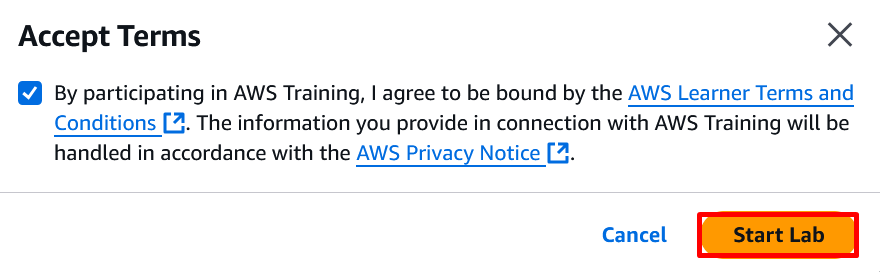
* Create an S3 bucket.
* Configure the S3 bucket as a static website and allow public access.
* Add a bucket policy.
* Create and upload the website assets.
* Test the Amazon S3 static website.



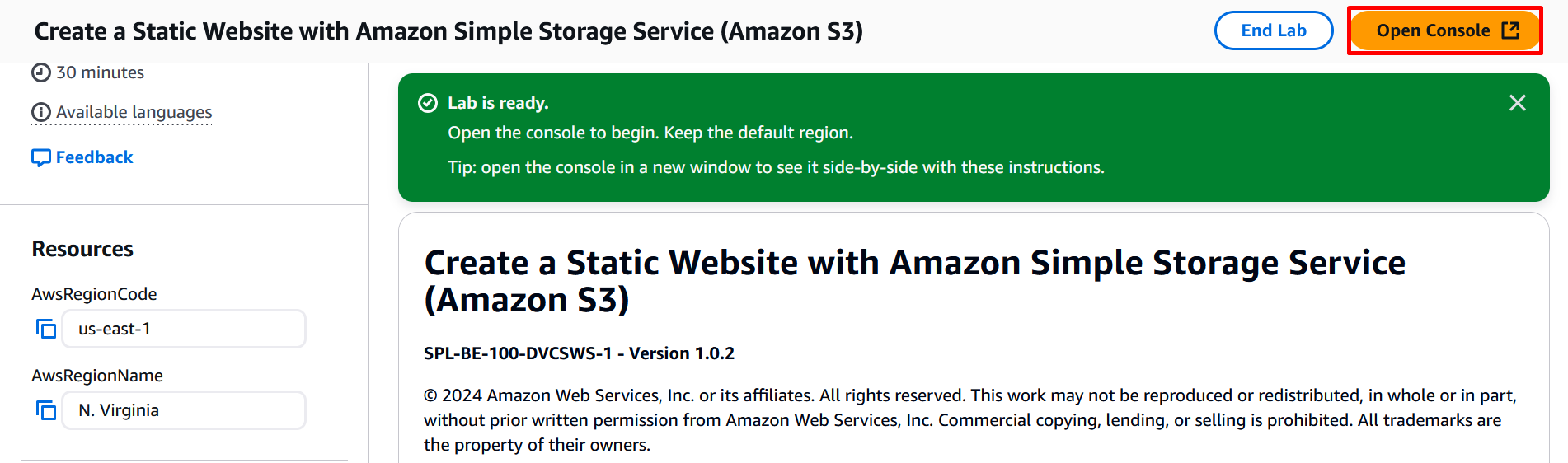
**Start lab**

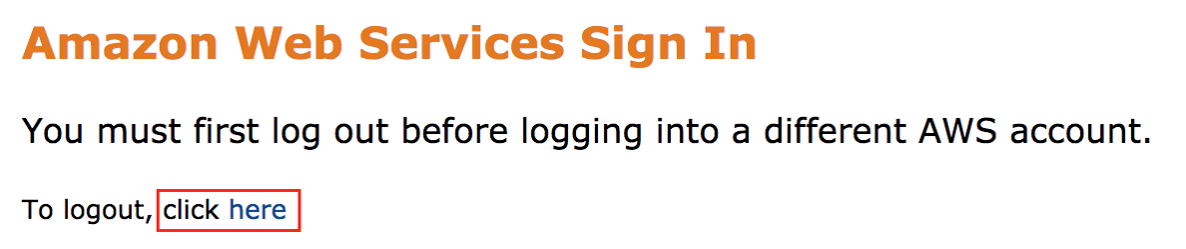
1. To launch the lab, at the top of the page, choose and click **Start lab**. We must wait for the provisioned AWS services to be ready before we can continue.





1. To open the lab, choose and click **Open Console**. We are automatically signed in to the AWS Management Console in a new web browser tab.





If we see the message, **we must first log out before logging into a different AWS account:**

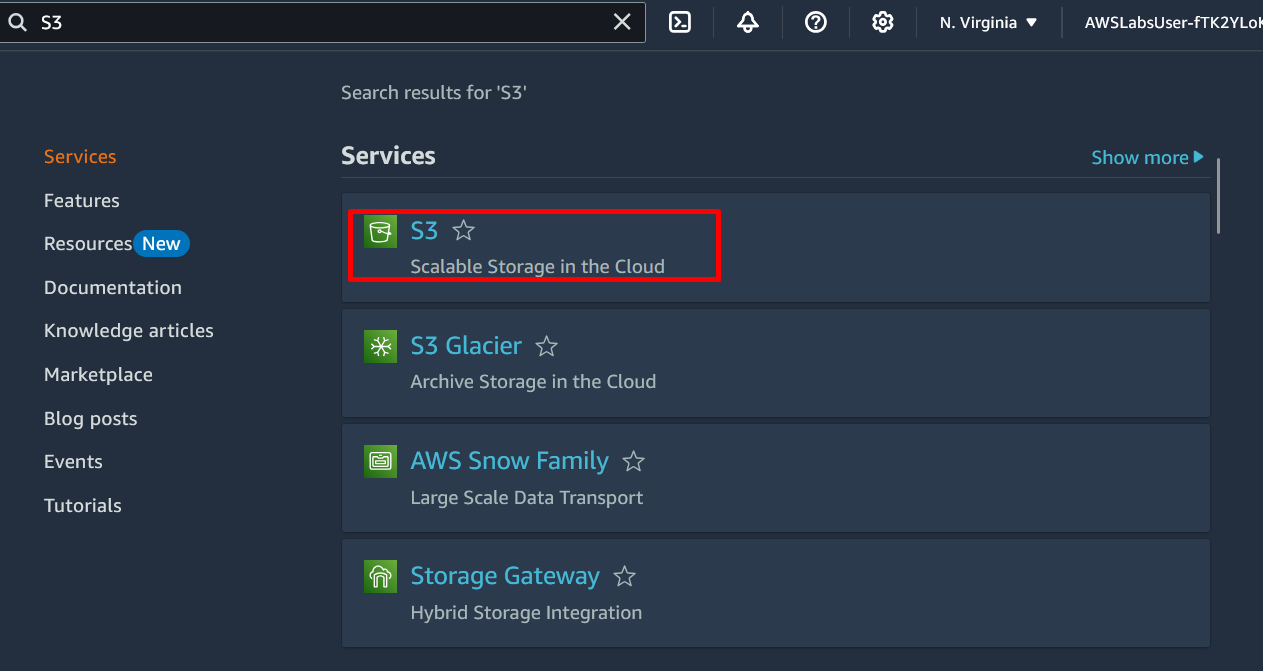
* Choose the **click here** link.
* Close our **Amazon Web Services Sign In** web browser tab and return to our initial lab page.
* Choose and click **Open Console** again.

**Task 1: Create an S3 bucket**

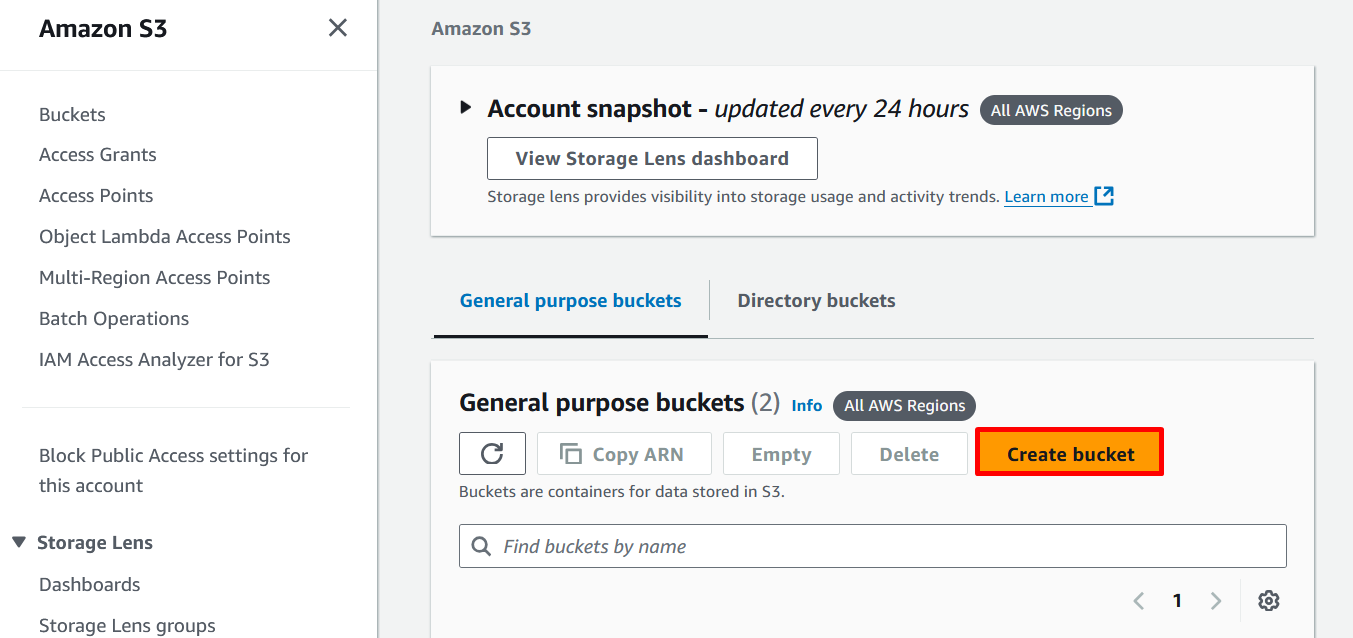
In this task, we create an Amazon S3 bucket. We can also set a number of properties. However, in this lab, we accept the default values so that we can see how to create a standard bucket.

* Be aware of the following naming considerations when we create an S3 bucket: The bucket name must be unique within a partition. A partition is a grouping of Regions. AWS currently has three partitions:
  + aws (Standard Regions),
  + aws-cn (China Regions), and
  + aws-us-gov (AWS GovCloud (US) Regions).
* It must be between **3** and **63** characters long.
* It must consist only of **lowercase letters, numbers, dots (.),** and **hyphens (-)**. It is recommend that we avoid using **dots (.)** in bucket names, except for buckets that are used only for static website hosting.
* It must begin and end with a **letter** or **number**.
* After we create the bucket, we can’t change its name.

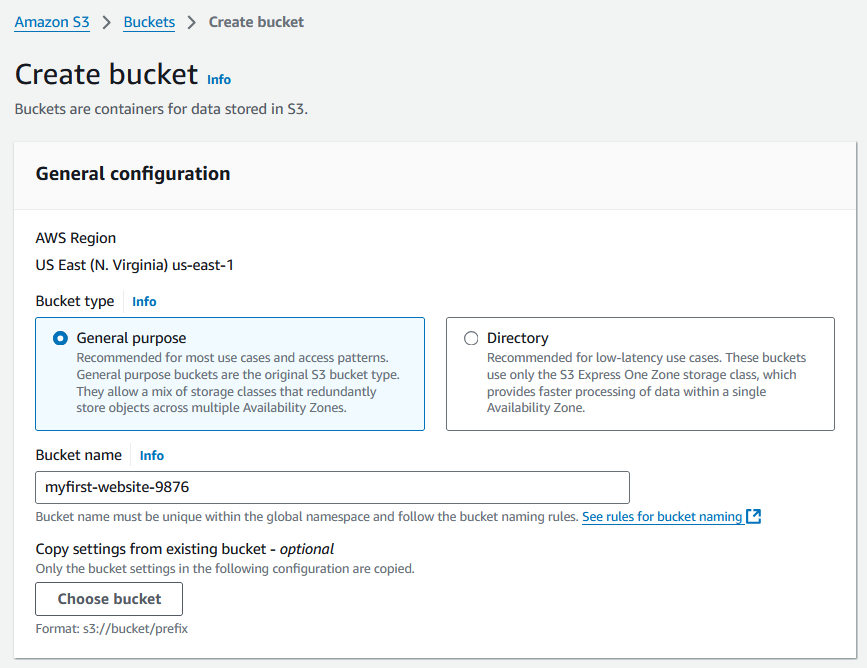
1. At the top of the AWS Management Console, in the search bar, search for and choose **S3**.



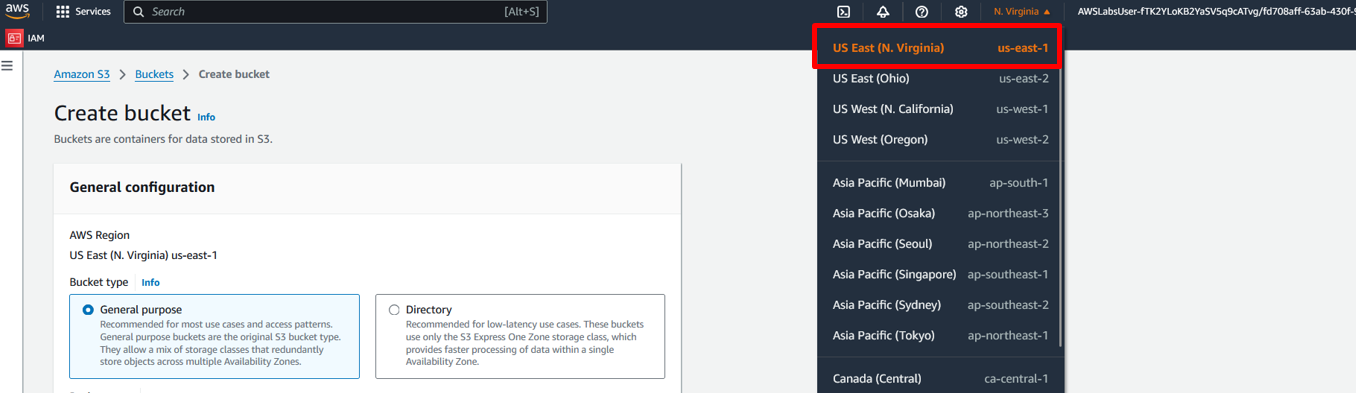
1. Choose and Click **Create bucket**.



1. For **Bucket name**, enter **ourInitials-website-ZipCode**  
   **Example:** mesh-website-75135

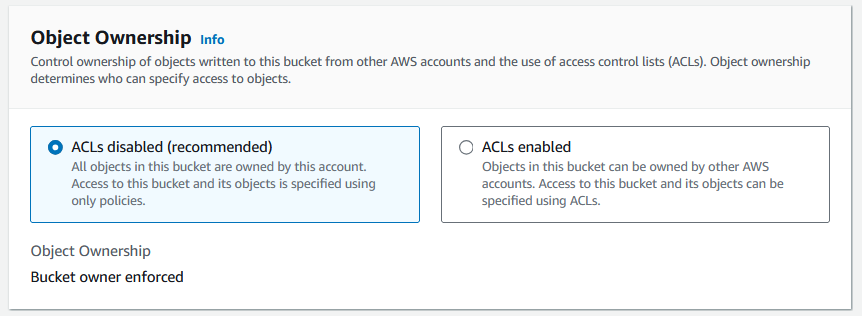


1. In the top right corner of the AWS console, confirm that the **AWS Region** matches the value to the left of these instructions, for example **N. Virginia**. If the region is different, choose the region name, and then select the correct region from the drop-down list.

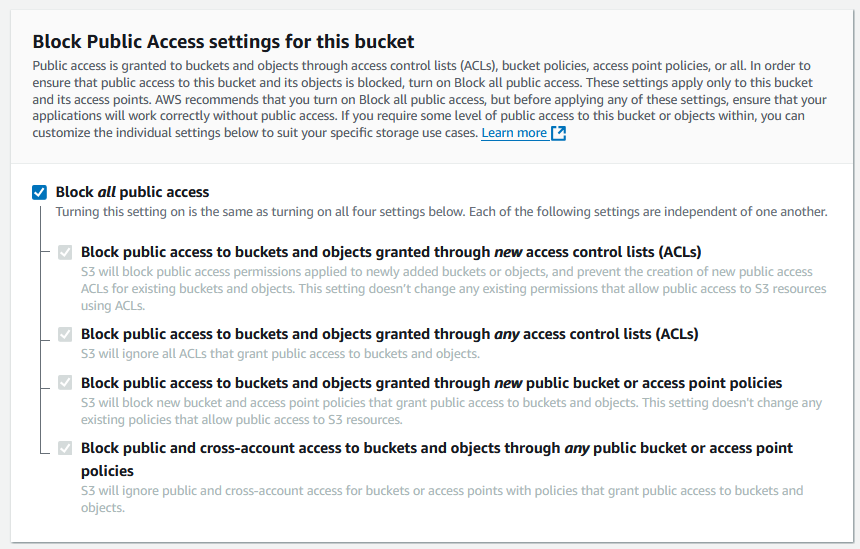


1. To create the bucket and accept the remaining default values for

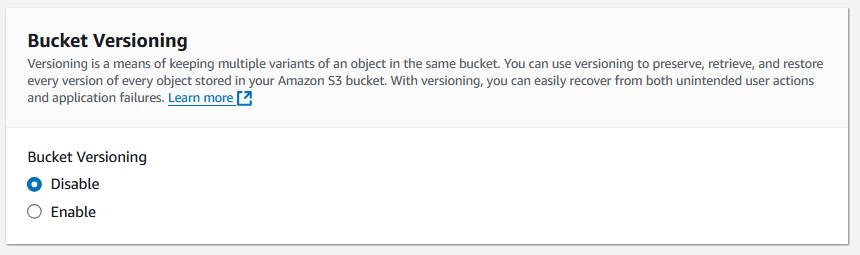
* **Object Ownership**,



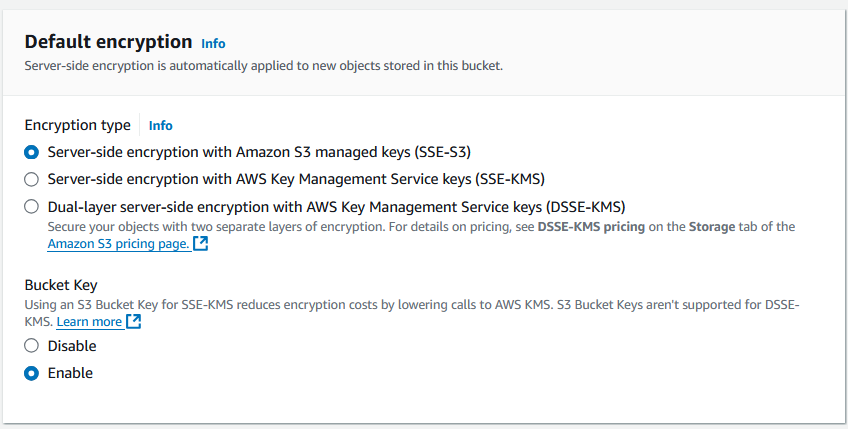
* **Block Public Access settings for this bucket**,



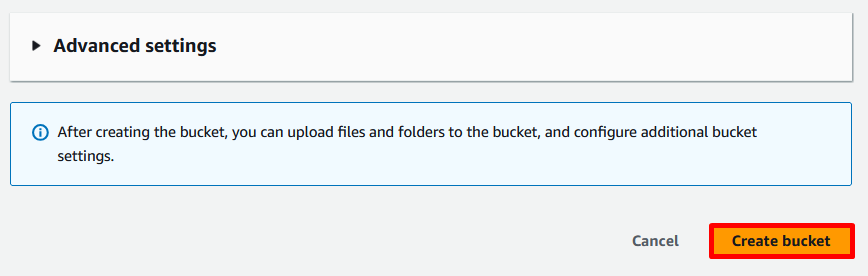
* **Bucket Versioning**,



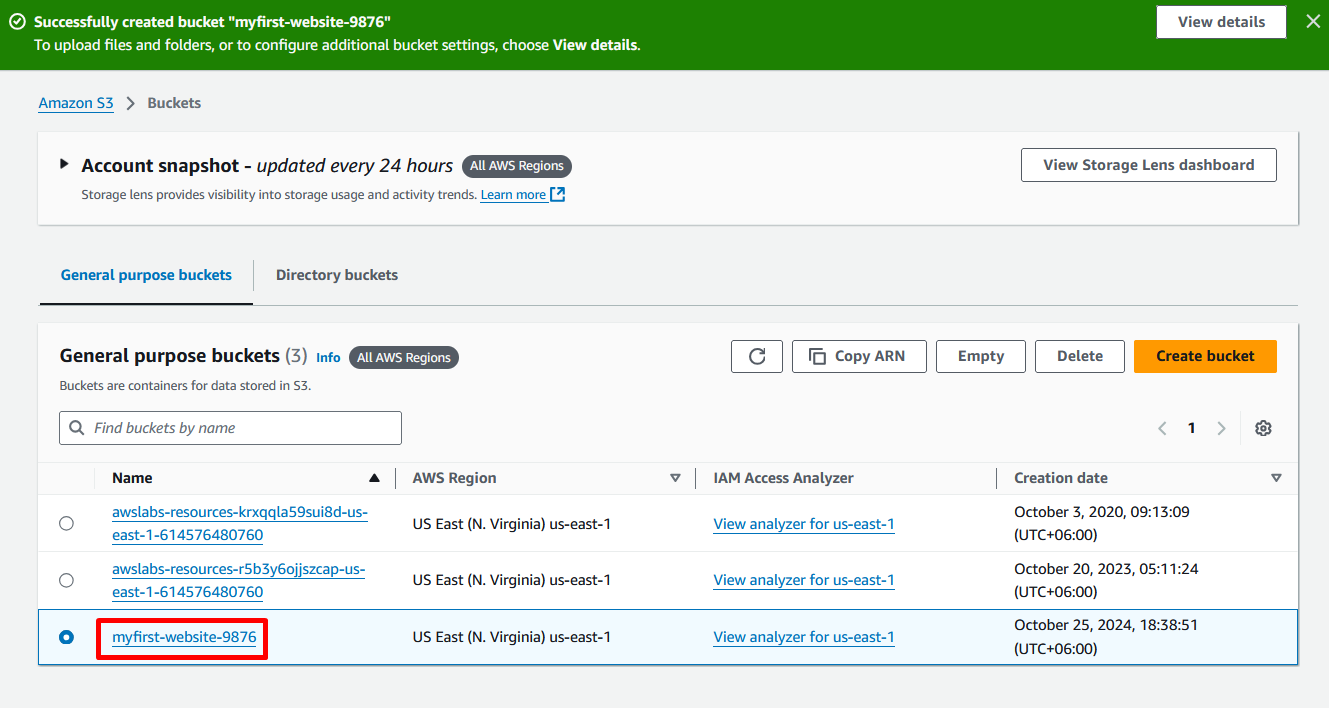
* And **Default Encryption**.



1. Choose and click **Create bucket** button.



A Successfully created bucket message is displayed on top of the screen. We have successfully created an S3 bucket to use as a static website.

  
  
**Noted:** By default, Amazon S3 now applies server-side encryption with Amazon S3 managed keys (SSE-S3) as the base level of encryption for every bucket in Amazon S3. With this encryption, all new object uploads to Amazon S3 are automatically encrypted at no additional cost and with virtually no impact on performance. Additionally, the buckets block public access unless we specifically choose to make them publicly available.

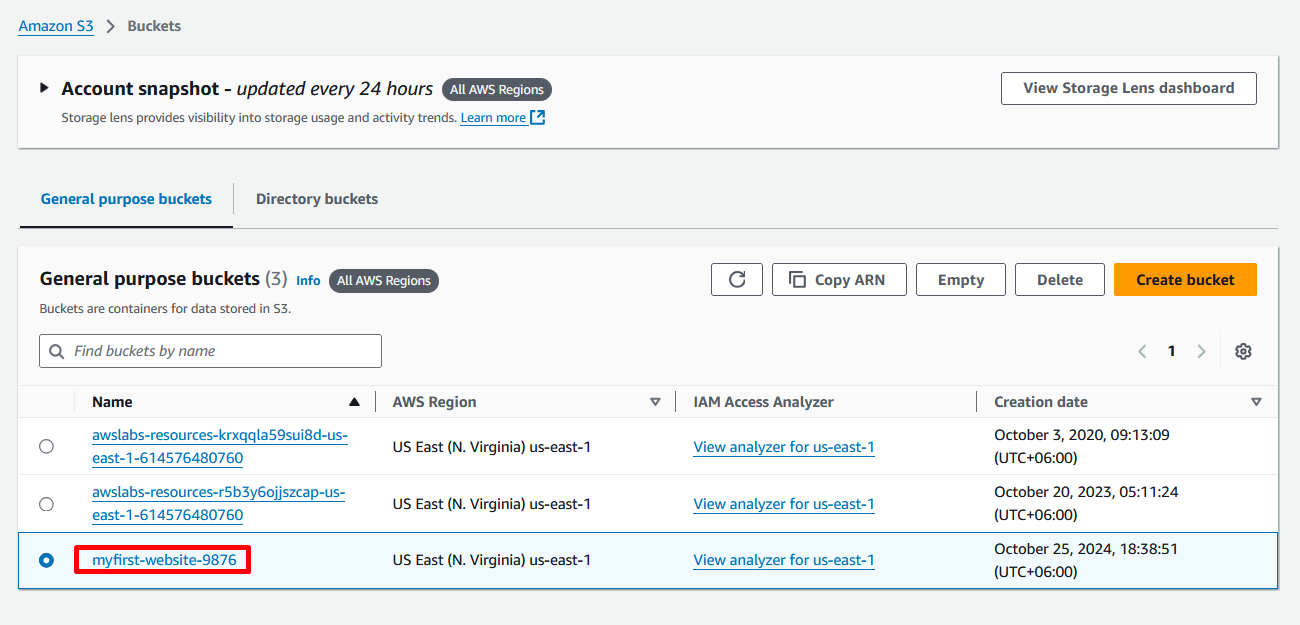
**Task 2: Configure the S3 bucket as a static website and allow public access**

In this task, we update the bucket’s properties to allow **static website hosting**. After we allow static website hosting, we also need to update the bucket’s permissions to allow public access to the bucket.

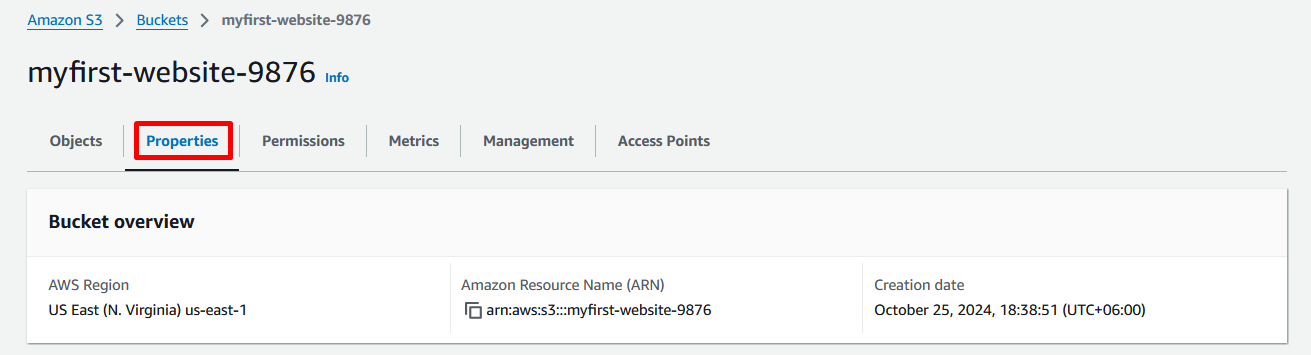
**Task 2.1: Configure the S3 bucket as a static website**

To use the bucket as a static website, we need to update the **Static website hosting** properties of the S3 bucket.

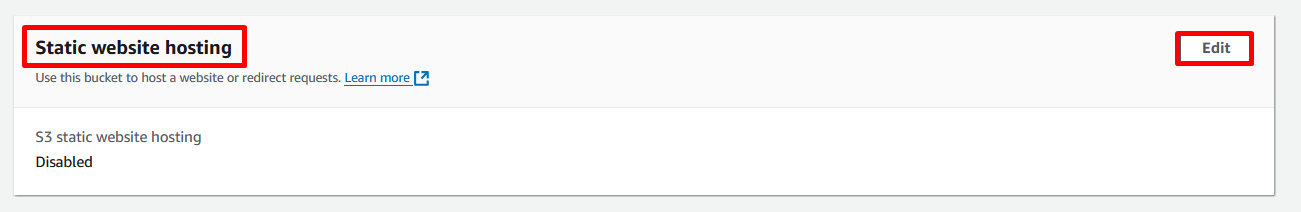
* + 1. Choose and click the **text link for the website bucket** that we just created.



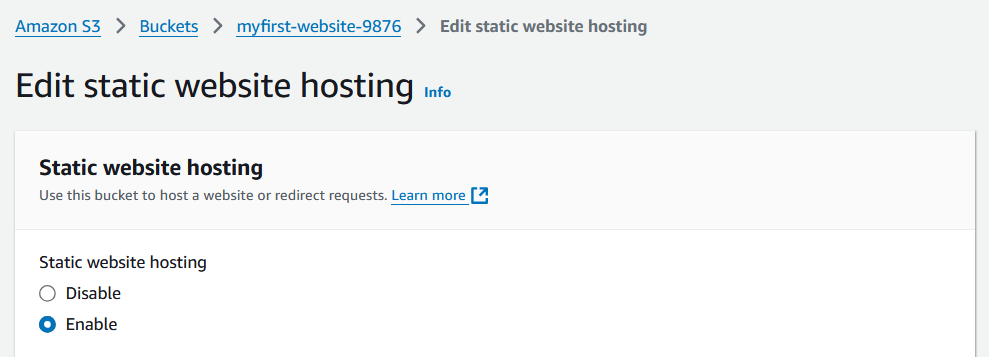
* + 1. In the **website bucket overview** page, choose and click the **Properties** tab.



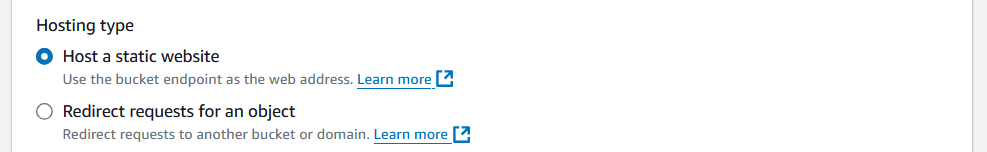
* + 1. Under **Static website hosting**, choose and click **Edit**. Then select the following options:



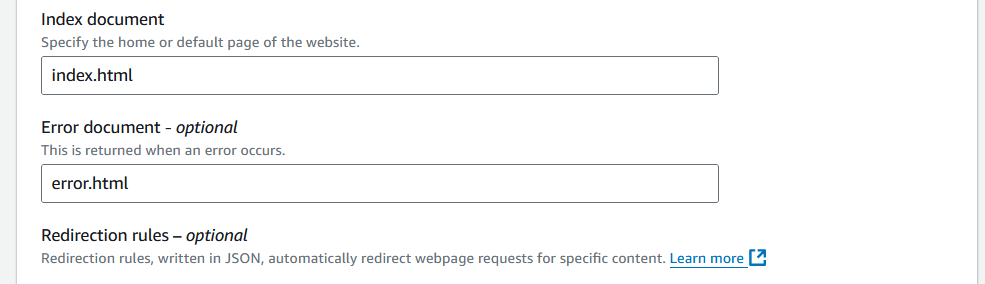
* For the **Static website hosting** option, select **Enable**.



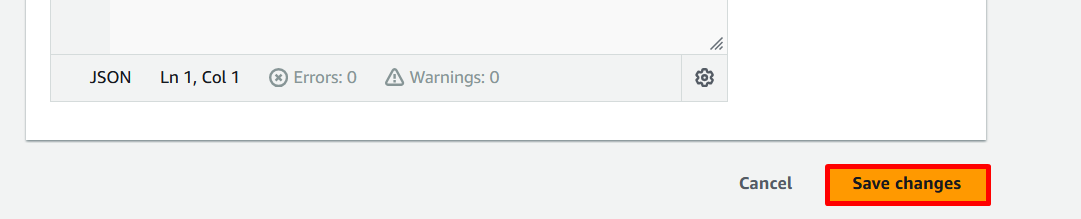
* For **Hosting type**, select **Host a static website**.



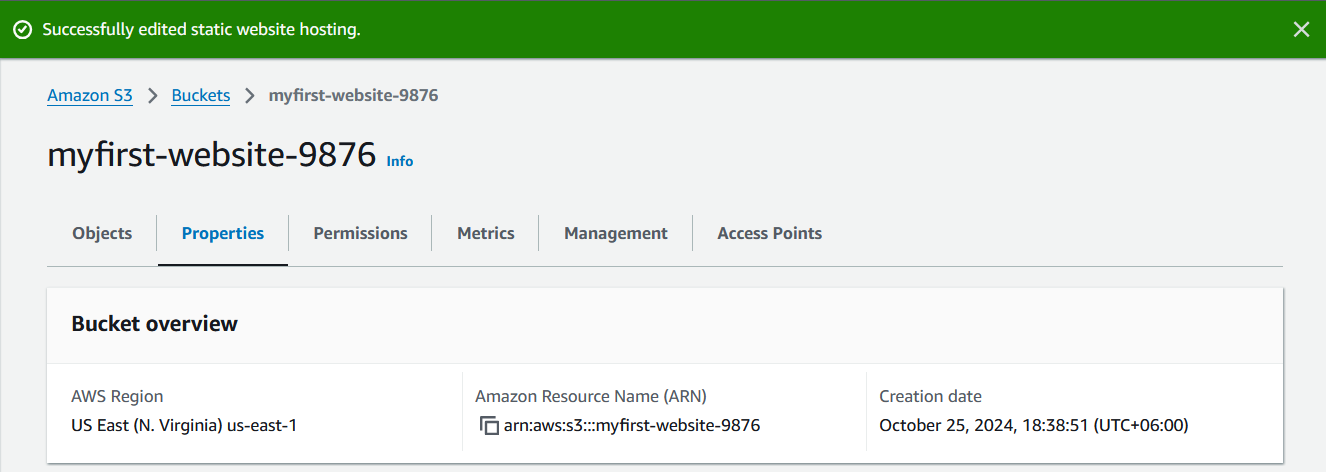
* For **Index document**, enter **index.html**
* For **Error document**, enter **error.html**



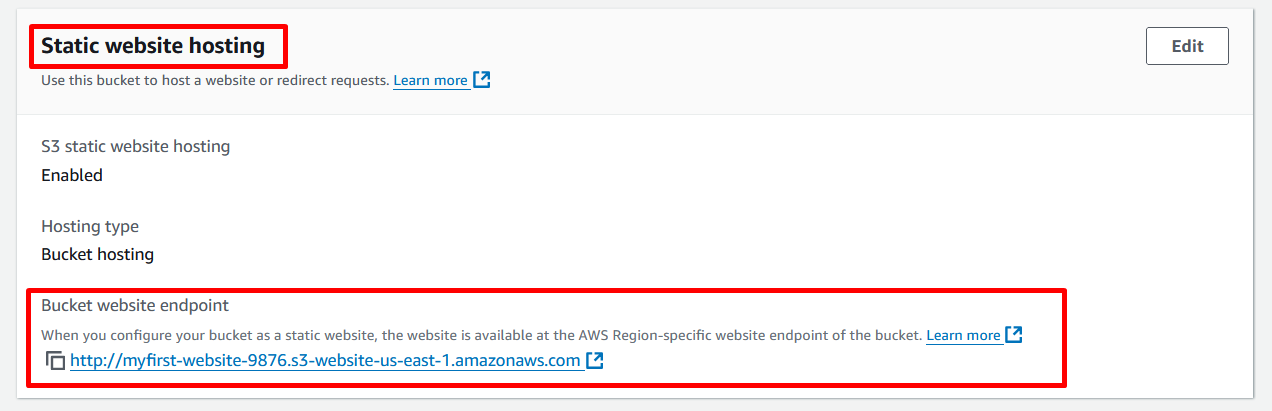
* + 1. Choose and click **Save changes** button.



A **Successfully edited static website hosting** message displays at the top of the window.



* + 1. Now, return to the **Static website hosting** section and then copy the value for the **Bucket website endpoint** and paste it into the web browser.

  
  
Now we see that, the following error page in the web browser.

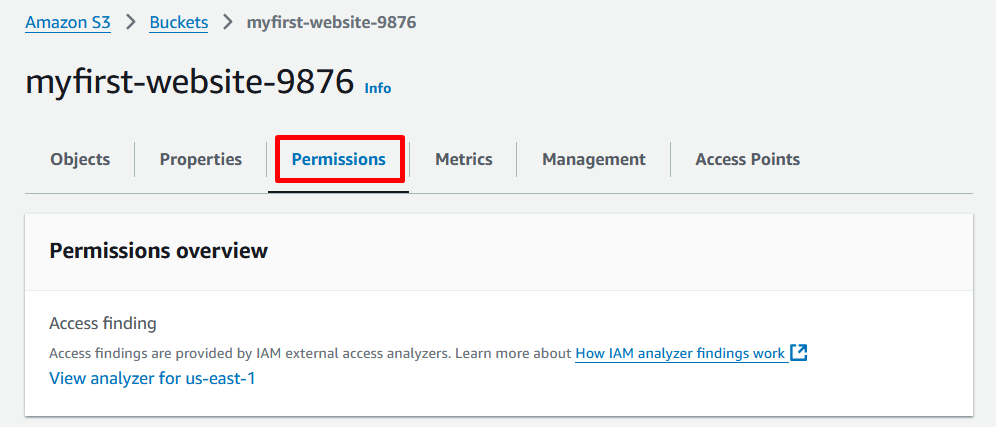


**Access Denied** will be displayed in the web browser. We need to configure a bucket policy to grant access to the bucket.

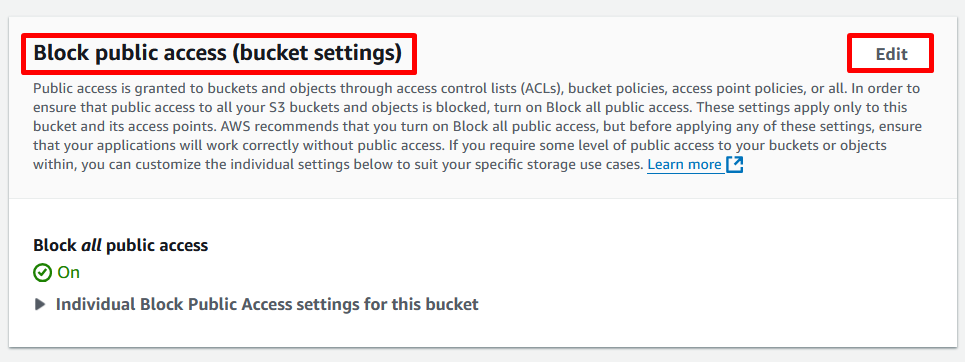
**Task 2.2: Allow public access to the bucket**

By default, Amazon S3 **blocks** public access to our account and buckets. If we want to use a bucket to host a static website, we must **allow** public access to the bucket. Use these steps to edit permissions for the **Block public access (bucket settings).**

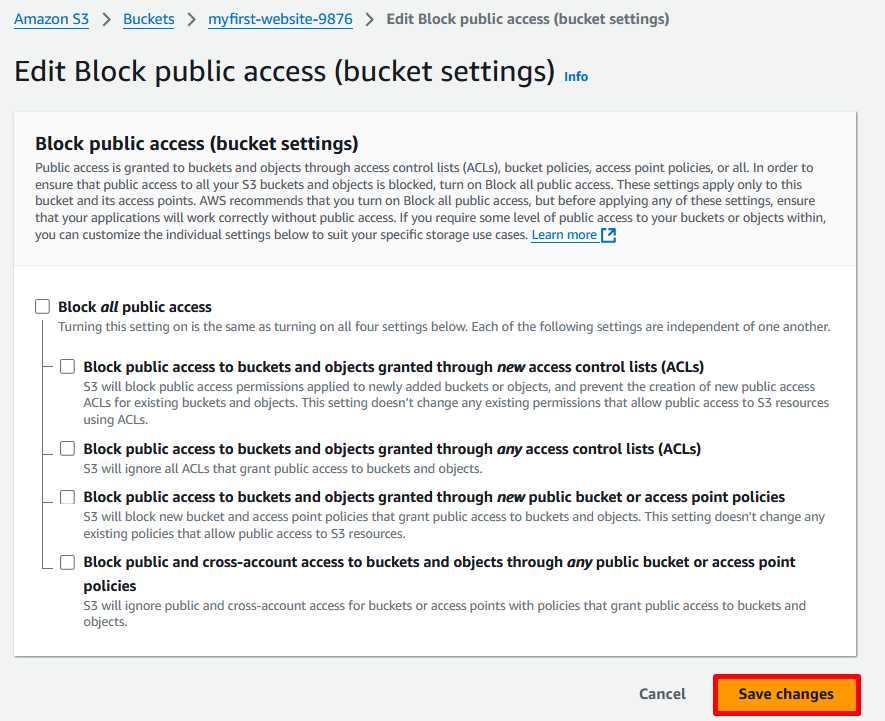
* + 1. In the **website bucket overview** page (which we already created), choose and click the **Permissions** tab.



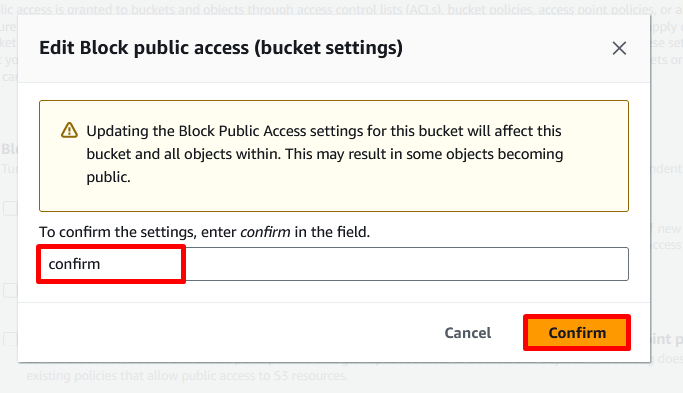
* + 1. From the **Block public access (bucket settings)** section, choose and click **Edit** button.



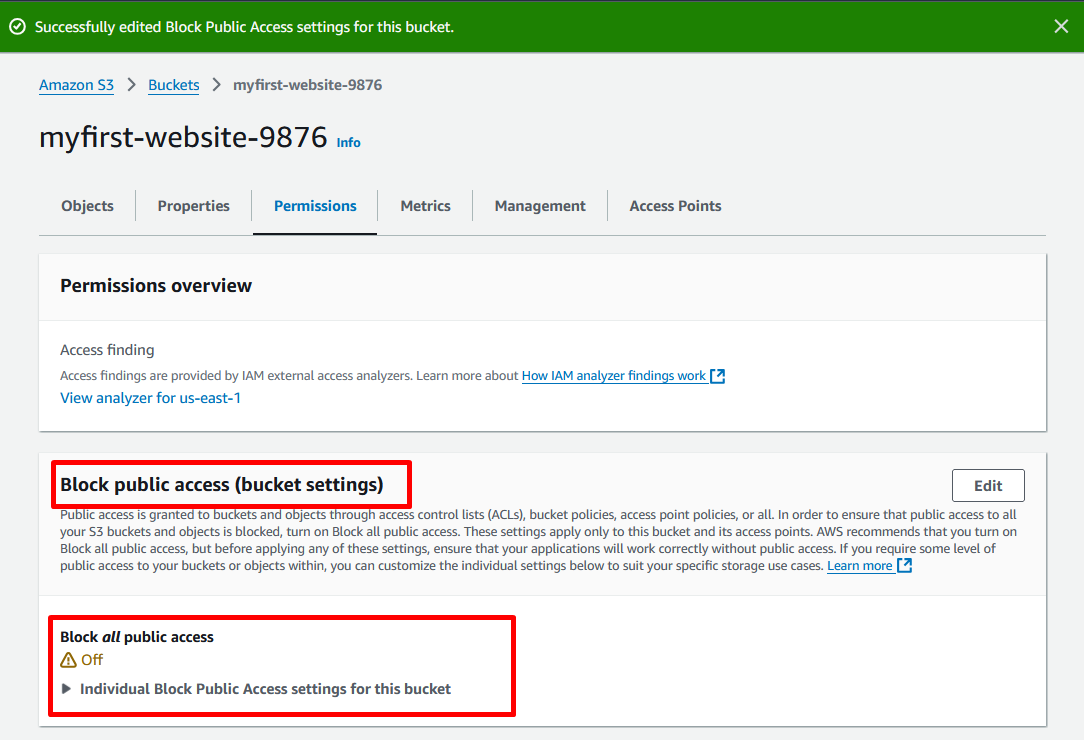
* + 1. Clear **Block** all **public access**, and choose and click **save changes** button.



* + 1. In the **Edit Block public access (bucket settings)?** Pop-up window, enter **confirm** and choose and click **Confirm** button.



A **Successfully edited Block public Access settings for the bucket** message displays at the top of the window.



We have successfully updated the bucket properties to allow static website hosting and updated permissions to allow public access to the bucket.

**Task 3: Add a bucket policy to allow public access to content in our bucket**

In this task, we have edited S3 **Block Public Access settings** to allow public access to the bucket and add a bucket policy to grant **public read access** to content in our bucket. When we grant public read access, anyone on the internet can access our bucket. We can accomplish this task by adding the **s3:GetObject** action to our S3 bucket and setting the **Principal** value to **\***, which represents anyone.

1. Choose and click the **Permissions** tab.

A screenshot of a computer

Description automatically generated

1. Under **Bucket Policy**, choose and click **Edit** button.

A screenshot of a computer

Description automatically generated

1. To grant **public read access** for our website, copy the following bucket policy and in the **Bucket policy editor**, paste the policy.

{

"Version": "2012-10-17",

"Statement": [

{

"Sid": "PublicReadGetObject",

"Effect": "Allow",

"Principal": "\*",

"Action": [

"s3:GetObject"

],

"Resource": [

"arn:aws:s3:::Bucket-Name/\*"

]

}

]

}

A screenshot of a computer

Description automatically generated

1. Update **Resource** to use **our** bucket name.

**Note:** In a bucket policy, the value of **Resource** is the Amazon Resource Name (ARN) for the bucket, where **Bucket-Name** is a placeholder for the name of our bucket. To use this bucket policy with our own bucket, we must update this name to match our **bucket name**. Be sure to include the trailing ***/*\*** characters because they are important for the functionality of the overall policy. For example, the bucket name in this example is **mesh-website-75135**.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* This is an EXAMPLE ONLY. \*\*\*\*

\*\*\*\* DO NOT COPY. \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

{

"Version": "2012-10-17",

"Statement": [

{

"Sid": "PublicReadGetObject",

"Effect": "Allow",

"Principal": "\*",

"Action": [

"s3:GetObject"

],

"Resource": [

"arn:aws:s3:::mesh-website-75135/\*"

]

}

]

}

A screenshot of a computer

Description automatically generated

1. After we update the policy, choose and click **Save changes** button.

A screenshot of a computer

Description automatically generated

A **Successfully edited bucket policy** message displays at the top of the window.

A screenshot of a computer

Description automatically generated

In this policy, we are granting **GetObject** permissions for our specific bucket to anyone.We have successfully added a bucket policy to our S3 bucket. This policy allows public access to the contents in our bucket.

**Task 4: Create and upload the website assets and test the website**

In this task, we create and upload the website assets to our S3 bucket. Website assets include images, style sheets, scripts, fonts, and other files that contribute to the visual and functional aspects of the website. In this lab, to help simplify the experience, the website assets include an **index.html** file and an **error.html** file. After we upload the assets to the S3 bucket, we test the website by opening a web browser to the Amazon S3.

**Task 4.1: Create the website assets**

The website assets for this lab include an **index.html** file and an **error.html** file, both that we create. The HTML code for the files is listed below. To create the files, copy the contents below, paste them into a text editor, and then save the files as **index.html** and **error.html** respectively. Once the two HTML files are created, we upload them to our bucket.

1. Create the **index.html** file. This file serves as the initial landing page for the website.
2. Open a text editor, create a new file.
3. In our text editor, Copy and paste the following HTML code.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Amazon S3 Static Website</title>

<style>

@font-face {

font-family: 'AmazonEmber';

src: url('https://m.media-amazon.com/images/G/01/AUIClients/AmazonUIFont-amazonember\_rg-cc7ebaa05a2cd3b02c0929ac0475a44ab30b7efa.\_V2\_.woff2') format('woff2');

}

body {

display: flex;

flex-direction: column;

justify-content: center;

align-items: center;

height: 100vh;

margin: 0;

font-family: Arial, sans-serif;

}

img {

max-width: 269px;

}

p {

position: absolute;

top: 350px;

text-align: center;

}

</style>

</head>

<body>

<h2>This is an Amazon S3 static website in the cloud!</h2>

<img src="https://d0.awsstatic.com/logos/powered-by-aws.png" alt="Powered by AWS Cloud Computing">

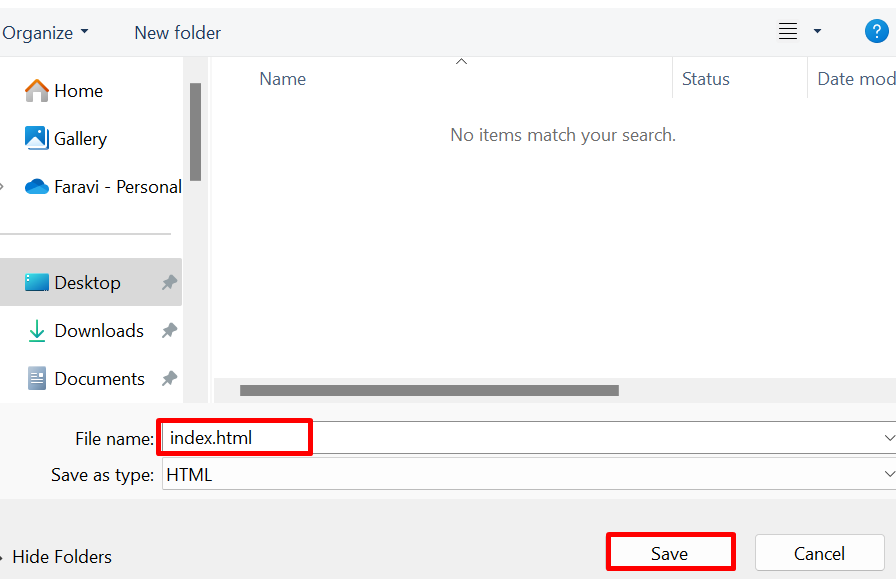
</body>

</html>

A screenshot of a computer program

Description automatically generated

1. Save the file as **index.html** to our desktop.



1. Next, we create the **error.html** file. This webpage is returned when a user enters an incorrect URL for this website.
2. In our text editor, create a new file.
3. Then, Copy and paste the following HTML code.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Error 404</title>

<style>

@font-face {

font-family: 'AmazonEmber';

src: url('https://m.media-amazon.com/images/G/01/AUIClients/AmazonUIFont-amazonember\_rg-cc7ebaa05a2cd3b02c0929ac0475a44ab30b7efa.\_V2\_.woff2') format('woff2');

}

body {

display: flex;

flex-direction: column;

justify-content: center;

align-items: center;

height: 100vh;

margin: 0;

font-family: Arial, sans-serif;

}

h3 {

text-align: center;

}

img {

max-width: 269px;

}

p {

position: absolute;

top: 350px;

text-align: center;

}

</style>

</head>

<body>

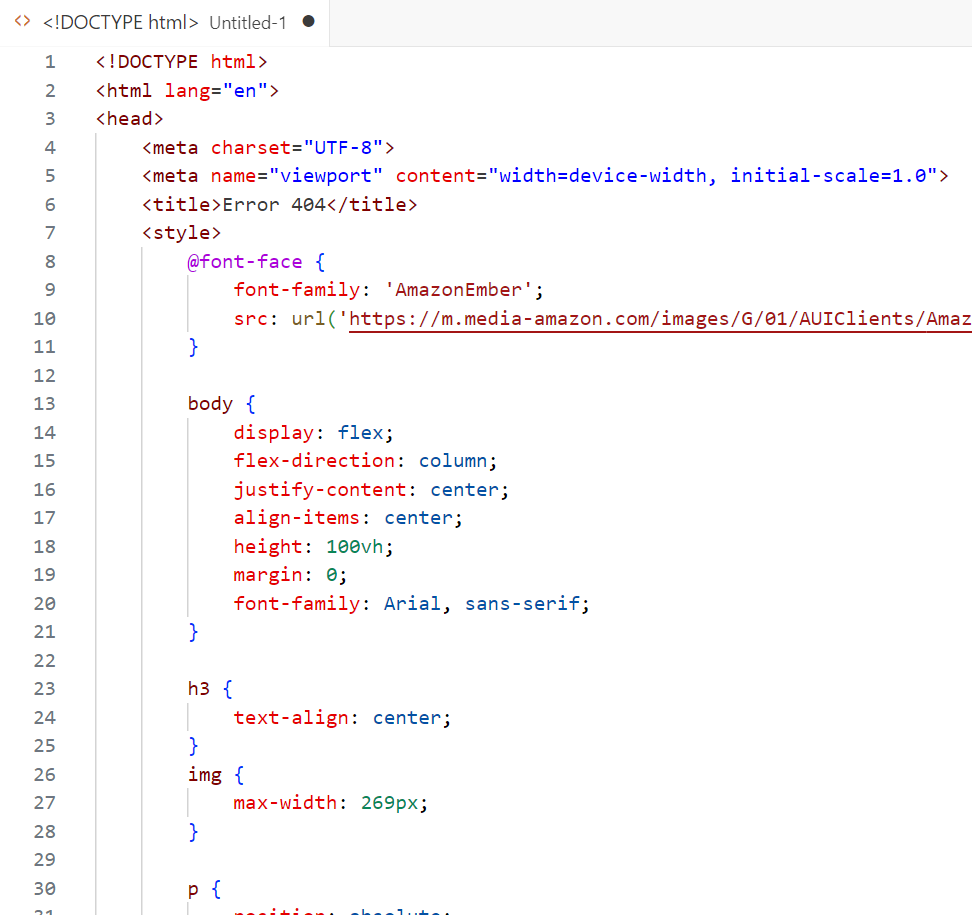
<h1>Error 404</h1>

<h3>Oops!<br><br>The URL we entered cannot be found.<br><br>Verify the URL and try again.</h3><br>

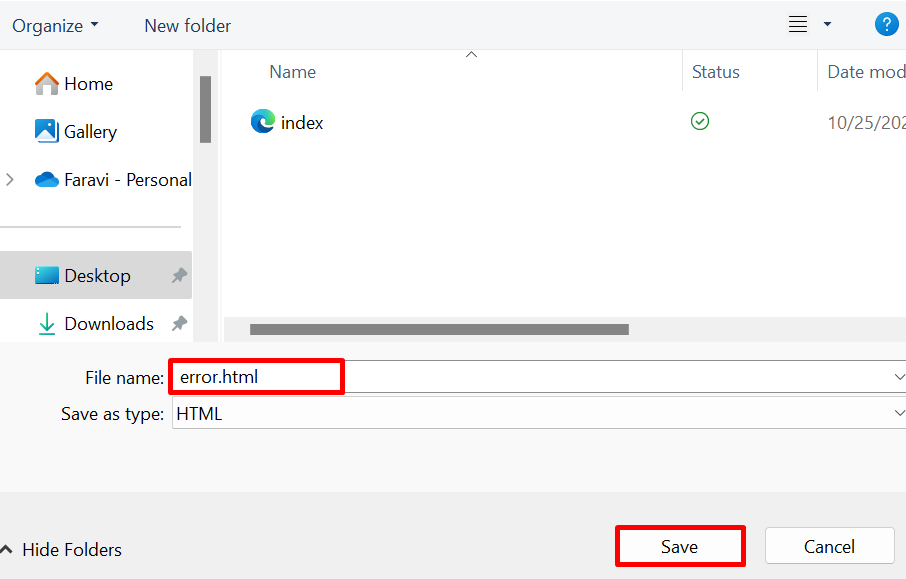
<img src="https://d0.awsstatic.com/logos/powered-by-aws.png" alt="Powered by AWS Cloud Computing">

</body>

</html>



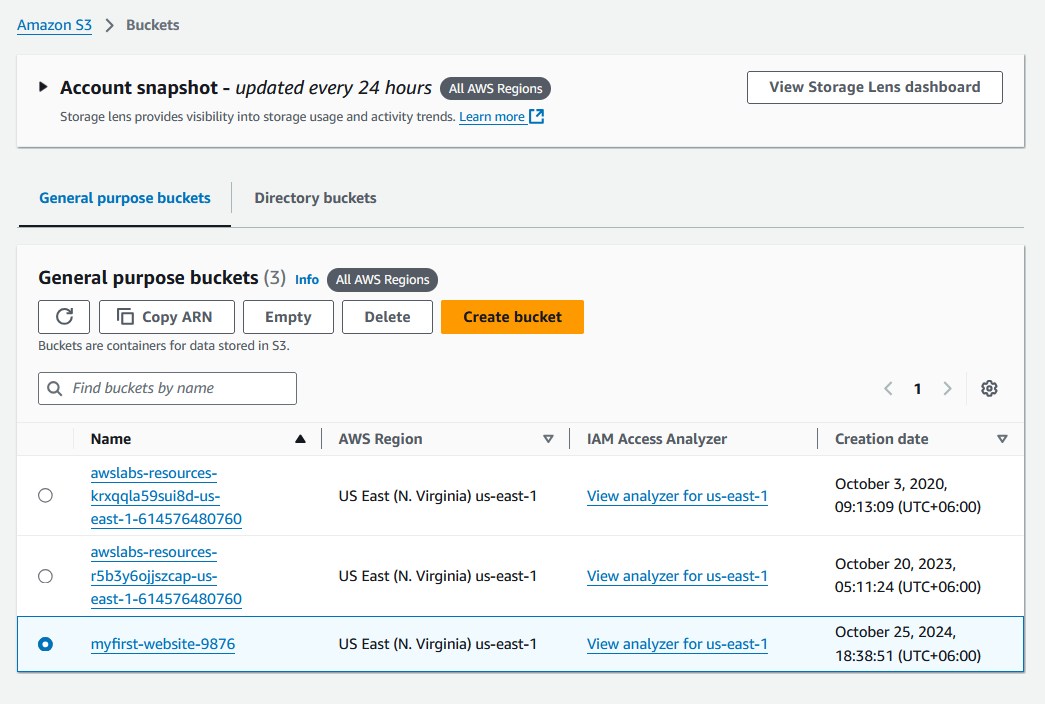
1. Save the file as **error.html** to our desktop.



**Task 4.2: Upload the website assets to our bucket**

Now that we have created the website assets, we can upload the assets to our bucket.

1. Return to the browser tab with the Amazon S3 **Bucket** List page. Choose and click **our** **website bucket** that we already created.

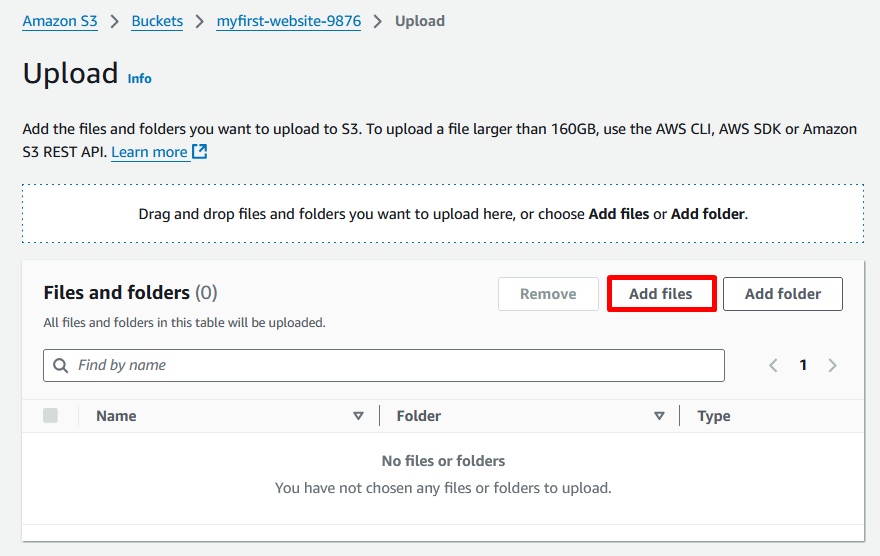


1. In the **website bucket overview** page, choose and click the **Objects** tab.Then, choose and click **Upload** button.

A screenshot of a computer

Description automatically generated

1. Choose and click **Add files** button.



1. Navigate to our desktop, select the **index.html** file and the **error.html** file, then choose and click **Open** button.

A screenshot of a computer

Description automatically generated

1. In the **Files and folders** section, we should see the following information about our files: their **names**, their **types**, and their **sizes**.

A screenshot of a computer

Description automatically generated

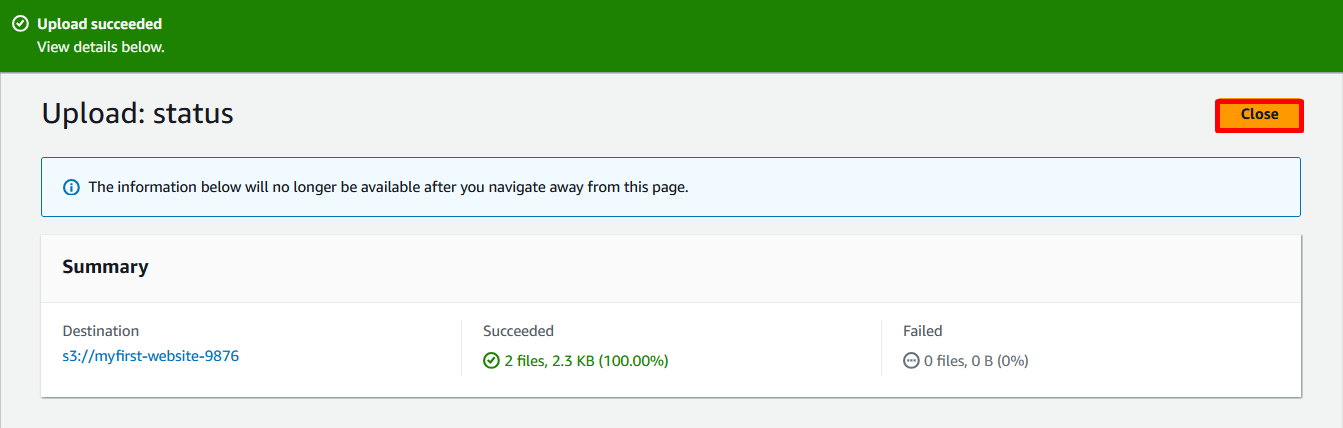
1. Choose and click **Upload** button.

A close-up of a person

Description automatically generated

An **Upload succeeded** message displays at the top of the window.

1. To return to the root of our bucket, choose and click **Close** button.



**Task 4.3: Test the Amazon S3 static website**

Now that we have uploaded the **index.html** and **error.html** files, it’s time to test the Amazon S3 static website.

1. Now again, return to the browser tab with the Amazon S3 **Bucket** List page. Choose and click **our** **website bucket** that we already created.
2. In the **website bucket overview** page, choose and click the **Properties** tab.
3. Then, go to the **Static website hosting** section, copy the value for the **Bucket website endpoint** and paste it into the new browser tab.
4. **Access the S3 URL.**

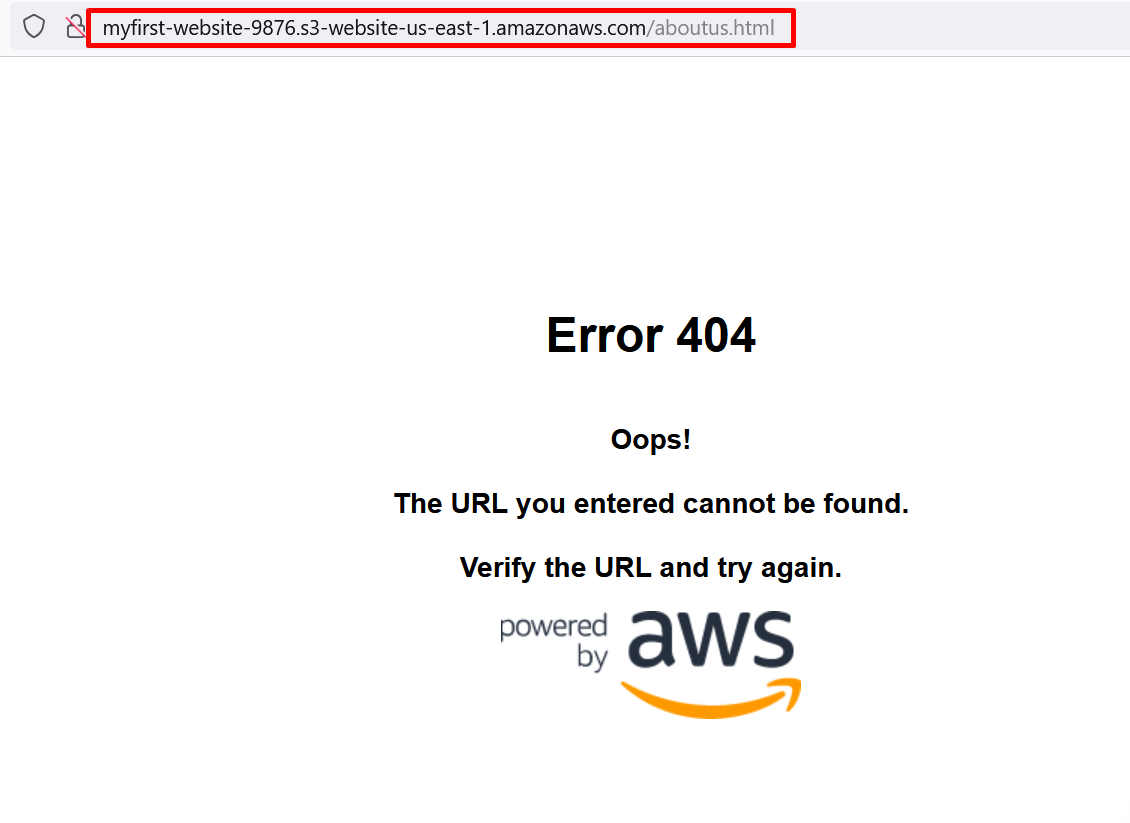
A screenshot of a computer

Description automatically generated  
  
Now, we see that the landing (**index.html**) page of the static website.

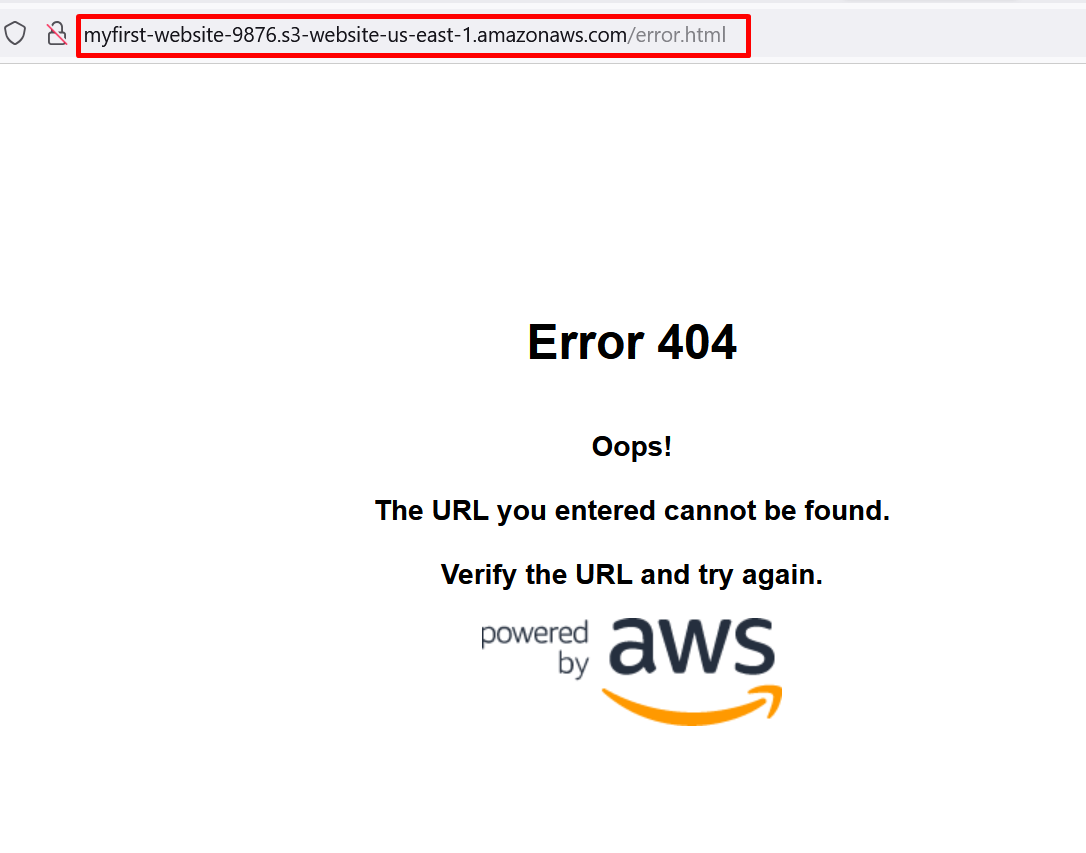
A screenshot of a computer

Description automatically generated

1. In the browser’s address bar, go to the end of the URL and add **/aboutus.html** to it, and press the **Enter** or **Return** key.



Now, we see that the error (**error.html**) page of the static website, when an incorrect URL is used.



**Warning:** The resources in this lab are created in an AWS account. When the lab terminates, those resources are automatically deleted. When we work in our own AWS account, we might build out various resources to learn from and test with as our journey with AWS evolves. For any resource that we don’t want or need to keep, remember to delete them so we don’t incur any unnecessary fees.

**Task complete:** We have successfully created both the **index.html** and **error.html** files. Then, we uploaded the files to our bucket. After the files were uploaded, we were able to view our Amazon S3 static website and the default error page.

**End lab**

Follow these steps to close the console and end our lab.

1. Return to the **AWS Management Console**.
2. At the upper-right corner of the page, choose and click **AWSLabsUser**, and then choose and click **Sign out**.
3. Choose and click **End lab** and then confirm that we want to end our lab.