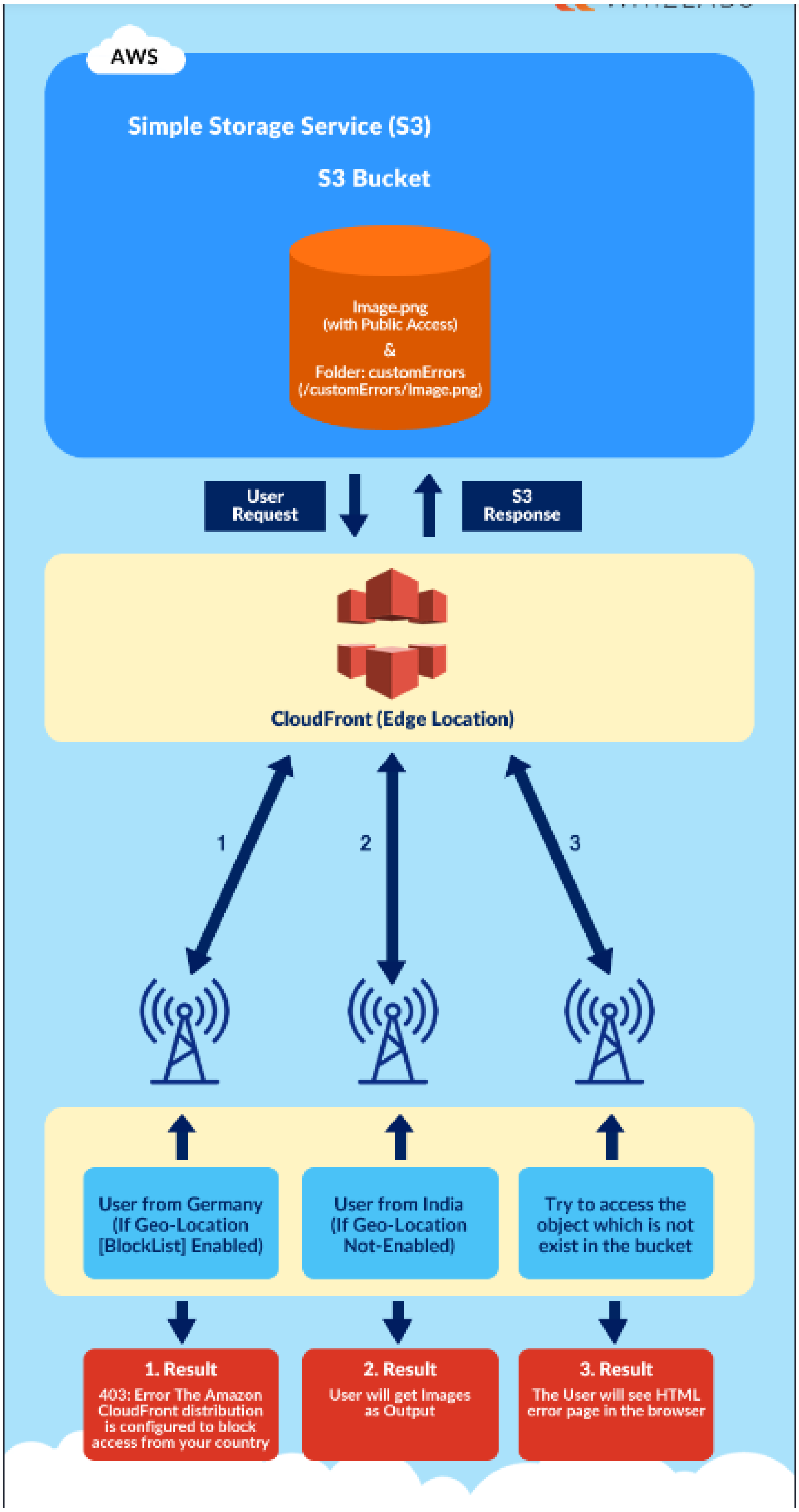
What is CloudFront?

1. Amazon CloudFront is a content delivery network (CDN) offered by AWS.
2. CDN provides a globally-distributed network of proxy servers which cache content , i.e., web videos or other bulky media, more locally to consumers, thus improving access speed for downloading the content.
3. CloudFront service works on a pay-as-you-go basis.
4. CloudFront works with origin servers like S3, EC2 where the content is stored and is pushed out to multiple CloudFront servers as content is requested.
5. When CloudFront is enabled, the content is stored on the main S3 server.
6. Copies of this content are created on a network of servers around the world called CDN.
7. Each server within this network is called an Edge server, which will only have a copy of your content.
8. When a request is made to the content, the user is provided from the nearest edge server.
9. CloudFront has features similar to dynamic site acceleration, a method used to improve online content delivery.

Architecture Diagram


AWS
Simple Storage Service (S3)
S3 Bucket
Image.png
(with Public Access)
&
Folder: customErrors
(/customErrors/Image.png)
User Request
S3 Response
CloudFront (Edge Location)
1
2
3
User from Germany (If Geo-Location [BlockList] Enabled)
User from India (If Geo-Location Not-Enabled)
Try to access the object which is not exist in the bucket
1. Result
403: Error The Amazon CloudFront distribution is configured to block access from your country
2. Result
User will get Images as Output
3. Result
The User will see HTML error page in the browser

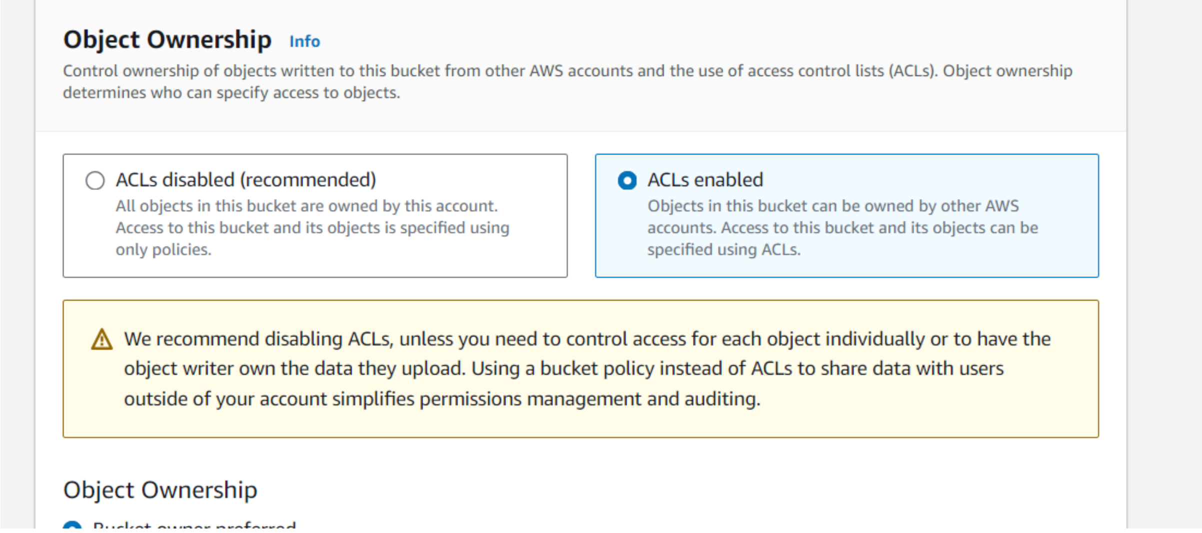
Task Details

1. Sign in to AWS Management Console
2. Create an S3 Bucket
3. Upload a file to the S3 bucket.
4. Create Custom Error pages.
5. Make the objects public.
6. Create a new Amazon CloudFront distribution.
7. Accessing images through Cloudfront.
8. Configuring custom Error Page
9. Restricting the Geographic Distribution of your content.

Task1: First sign in to AWS Management Console

Task2: Go and create an S3 Bucket

2.1 Under Object Ownership select ACL enabled



Object Ownership Info

Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

○ ACLs disabled (recommended)
All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

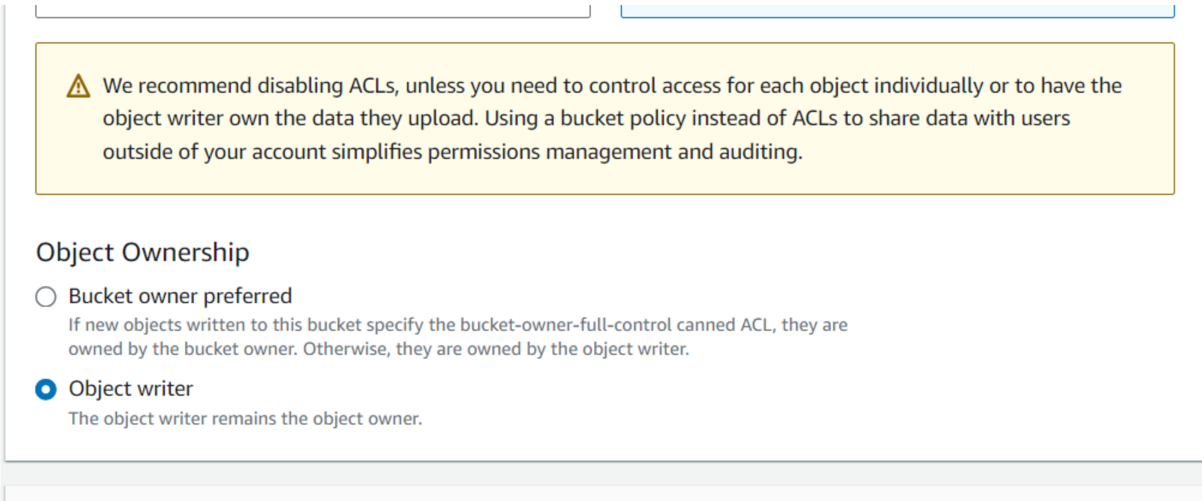
● ACLs enabled
Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

⚠ We recommend disabling ACLs, unless you need to control access for each object individually or to have the object writer own the data they upload. Using a bucket policy instead of ACLs to share data with users outside of your account simplifies permissions management and auditing.

Object Ownership

● Bucket owner preferred

2.2 Choose Object Writer



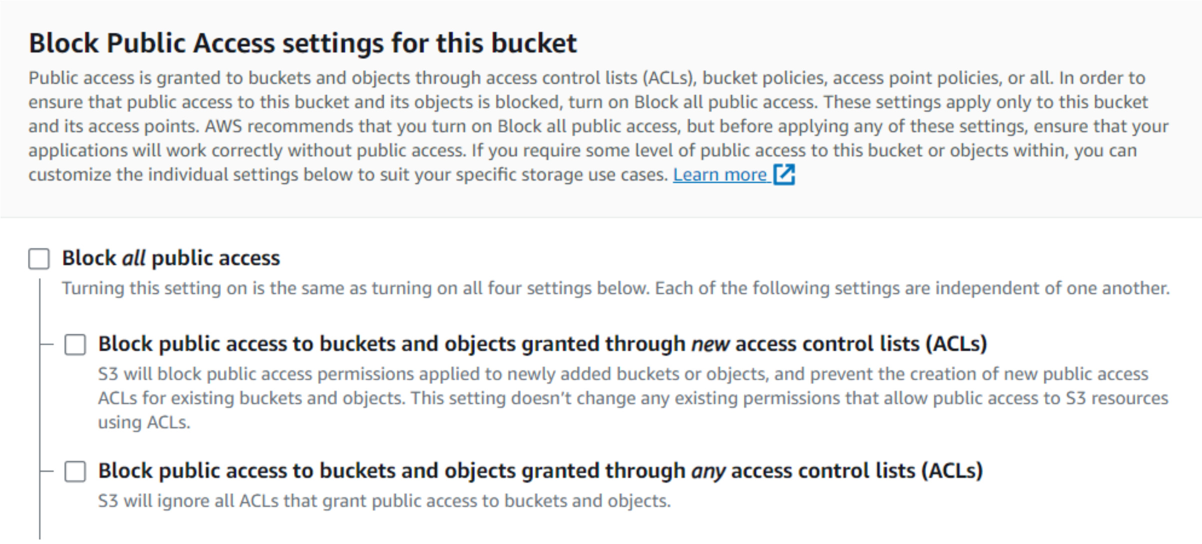
We recommend disabling ACLs, unless you need to control access for each object individually or to have the object writer own the data they upload. Using a bucket policy instead of ACLs to share data with users outside of your account simplifies permissions management and auditing.

Object Ownership

○ Bucket owner preferred
If new objects written to this bucket specify the bucket-owner-full-control canned ACL, they are owned by the bucket owner. Otherwise, they are owned by the object writer.

● Object writer
The object writer remains the object owner.

2.3 Uncheck Block all public access



Block Public Access settings for this bucket

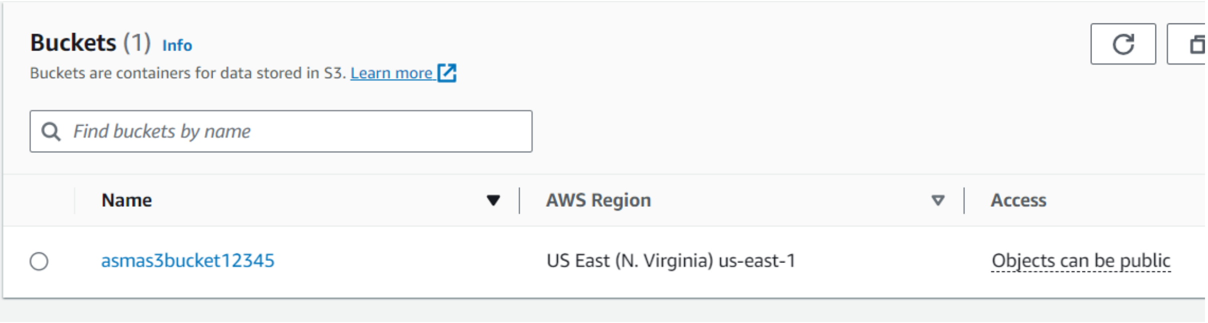
Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. Learn more

□ Block *all* public access
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

□ Block public access to buckets and objects granted through *new* access control lists (ACLs)
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.

□ Block public access to buckets and objects granted through *any* access control lists (ACLs)
S3 will ignore all ACLs that grant public access to buckets and objects.

2.4 Create the bucket.



Buckets (1) Info

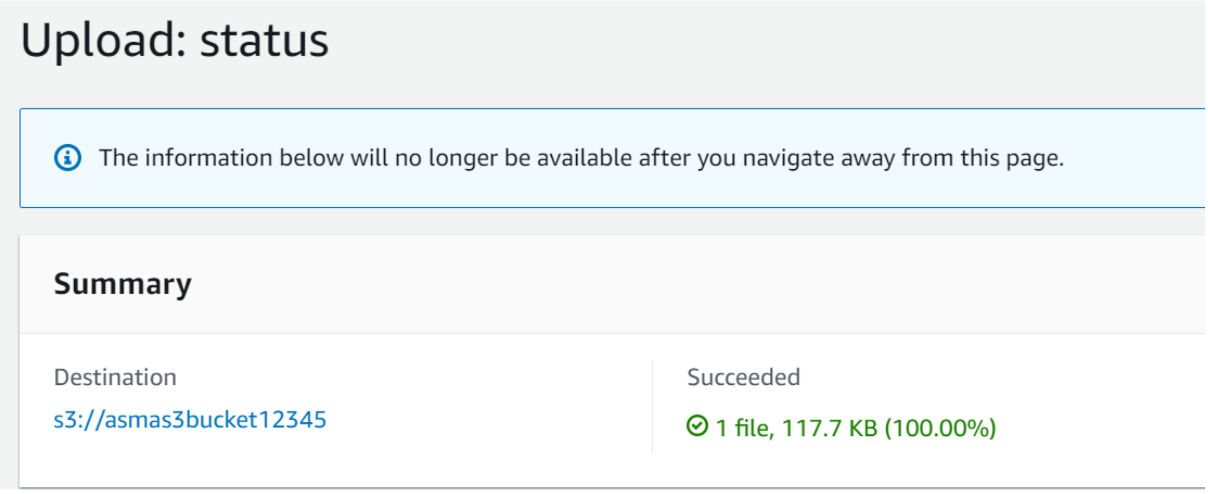
Buckets are containers for data stored in S3. Learn more

Find buckets by name

	Name	AWS Region	Access
○	asmas3bucket12345	US East (N. Virginia) us-east-1	Objects can be public

1. After creating successfully a Bucket, upload an image file to the S3 bucket.
   - Click on the **Upload** button.

- Click on **Add files**.
- Add the local image from the computer.
- Click on the **Upload** button.



Upload: status

ⓘ The information below will no longer be available after you navigate away from this page.

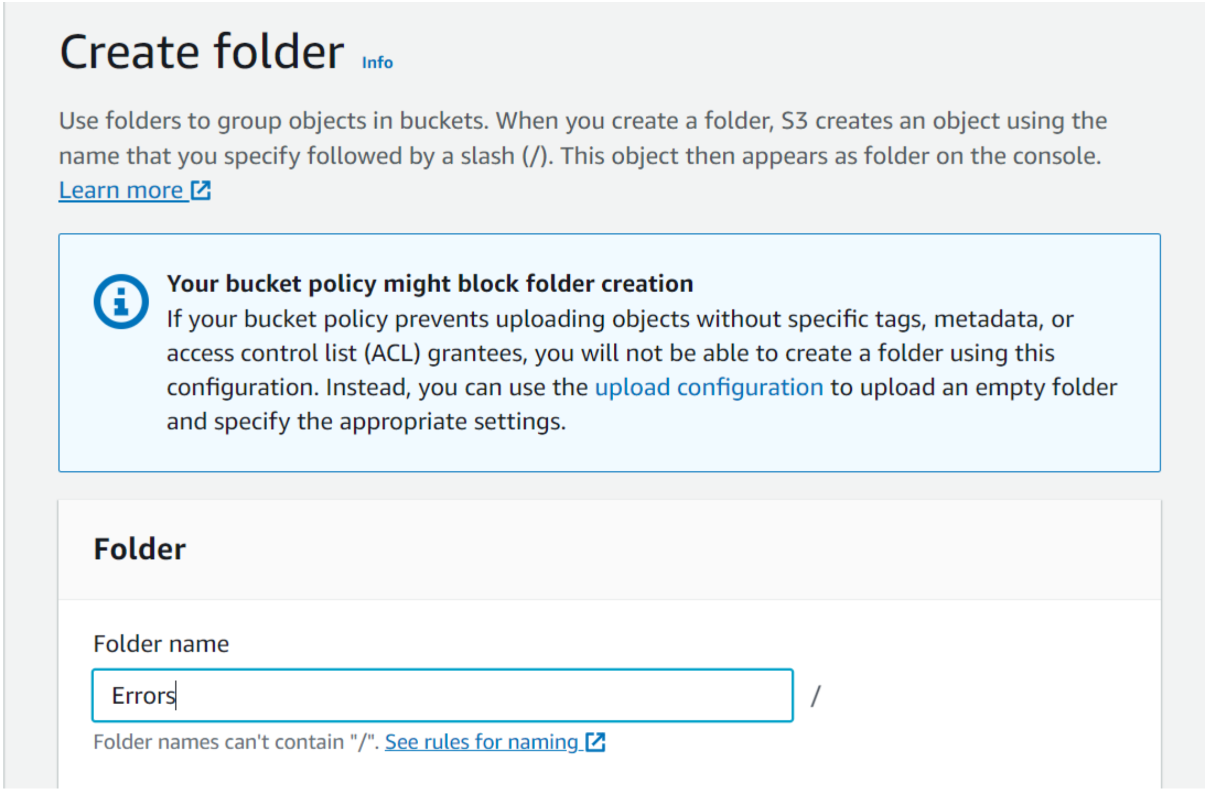
Summary

Destination
s3://asmas3bucket12345

Succeeded
✓ 1 file, 117.7 KB (100.00%)

1. Now we will learn to create customized error pages for CloudFront. These pages will be displayed when an origin returns an HTTP 4xx or 5xx error. For this, we must ensure that the error pages are stored in a location that CloudFront can access. In this case, we will use the same S3 bucket that we created previously.

4.1 Open the



Create folder Info

Use folders to group objects in buckets. When you create a folder, S3 creates an object using the name that you specify followed by a slash (/). This object then appears as folder on the console. Learn more

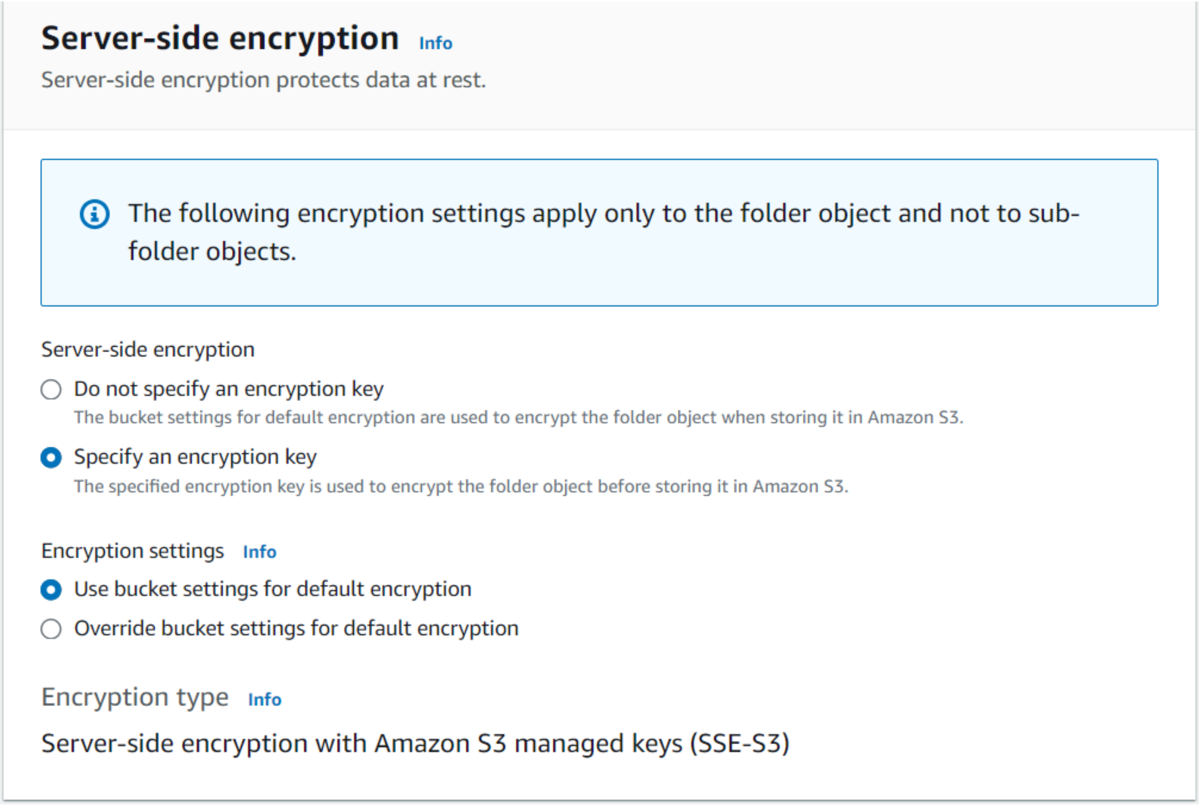
Your bucket policy might block folder creation
If your bucket policy prevents uploading objects without specific tags, metadata, or access control list (ACL) grantees, you will not be able to create a folder using this configuration. Instead, you can use the upload configuration to upload an empty folder and specify the appropriate settings.

Folder

Folder name

Errors /

Folder names can't contain "/". See rules for naming



Server-side encryption Info

Server-side encryption protects data at rest.

The following encryption settings apply only to the folder object and not to sub-folder objects.

Server-side encryption

○ Do not specify an encryption key
The bucket settings for default encryption are used to encrypt the folder object when storing it in Amazon S3.

● Specify an encryption key
The specified encryption key is used to encrypt the folder object before storing it in Amazon S3.

Encryption settings Info

● Use bucket settings for default encryption

○ Override bucket settings for default encryption

Encryption type Info

Server-side encryption with Amazon S3 managed keys (SSE-S3)

1. Click on the new **CustomErrors** folder.
2. We will create an **error.html** file:
   - Create an **error.html** file in your local system using Notepad.

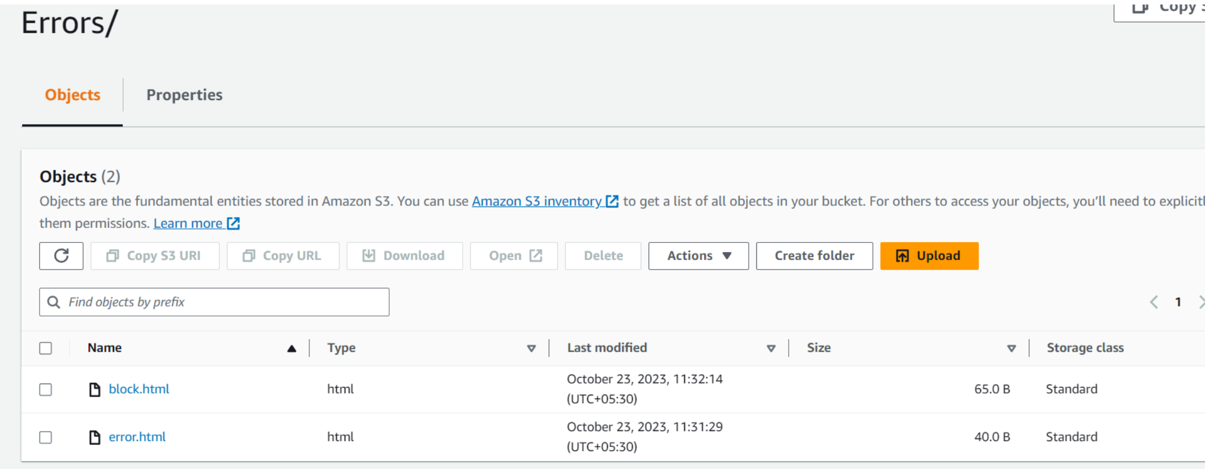
    - This custom HTML page will be used for showing errors in CloudFront.
    - Sample **error.html** content:

3. <html><h1>This is Error Page</h1></html>
4. Create a Create Custom Error pages.

- <html><h1>This is Error Page</h1></html>

1. Use the **Upload** button to upload the **error.html** file in the folder.
2. We will create a **block.html** file:
    - Create a **block.html** file in your local using Notepad.
    - This custom HTML page will be used for showing geo-restrictions of your content in CloudFront.
    - Sample **block.html** content:

<html><h1>This content is blocked in your location!!!</h1></html>



Objects (2)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitl them permissions. Learn more

□	Name	Type	Last modified	Size	Storage class
□	block.html	html	October 23, 2023, 11:32:14 (UTC+05:30)	65.0 B	Standard
□	error.html	html	October 23, 2023, 11:31:29 (UTC+05:30)	40.0 B	Standard

Task 3: Making the objects public

1. Click on the image name. You can see the image details like Owner, size, link, etc.
2. Copy the Object URL and paste it into a new tab.
3. A sample **Object URL**: https://asmas3bucket12345.s3.amazonaws.com/most-beautiful-gardens-in-the-world.jpg
    - You will see the **AccessDenied** message, meaning the object is not publicly accessible.most-beautiful-gardens-in-the-world.jpg.
    - Go back to the Bucket and click the **Permissions** tab.

5. Scroll down to the **Bucket Policy** and click on the **Edit** button.

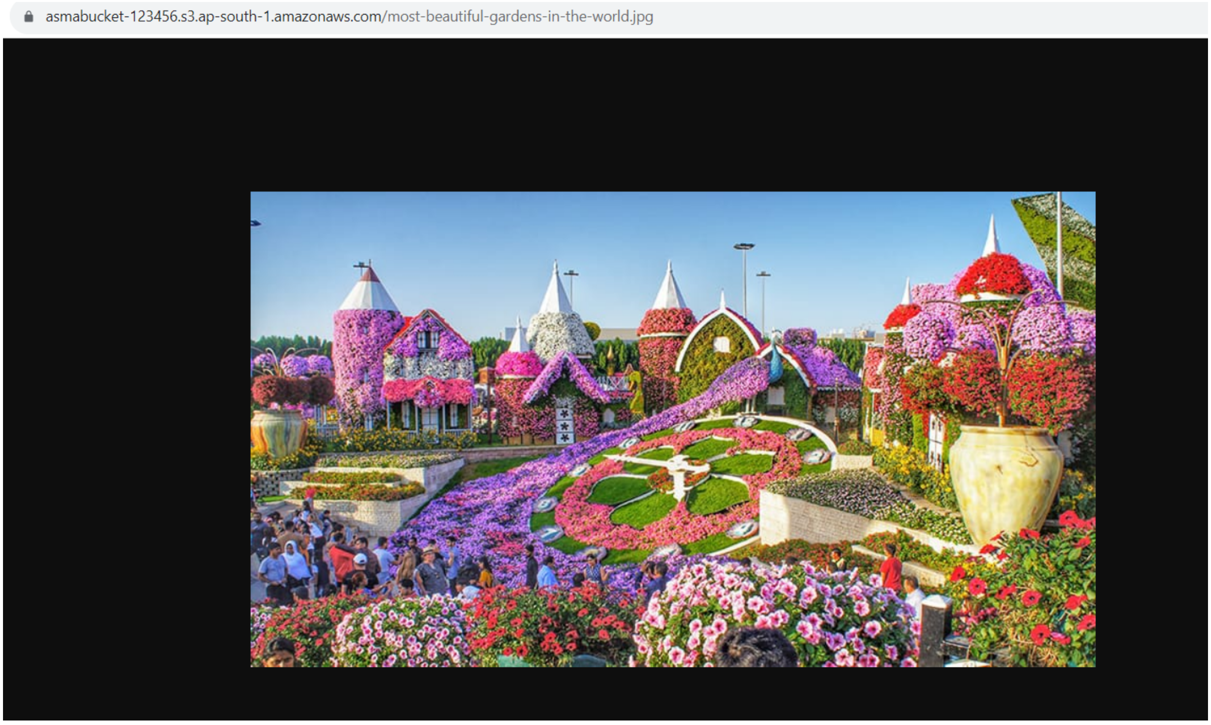
6. **Copy and paste** the below policy and save the policy.

- **Note**: Change the **name** of the **bucket ARN** with your **bucket ARN** in both the **Resource** option in the code.

```
{
"Version": "2012-10-17",
"Statement": [
{
"Effect": "Allow",
"Action": ["s3:ListBucket"],
"Principal": {"AWS": "*"},
"Resource": "<YOUR_BUCKET_ARN>"
},
{
"Effect": "Allow",
"Action": ["s3:GetObject", "s3:PutObject"],
"Principal": {"AWS": "*"},
"Resource": "<YOUR_BUCKET_ARN>/*"
}
]
}
```

7. Open the Image **Object URL** again or refresh the one already open.

8. If you can see your uploaded image in the browser, it means your image is publicly accessible. If not, check your bucket policy again.



asmabucket-123456.s3.ap-south-1.amazonaws.com/most-beautiful-gardens-in-the-world.jpg

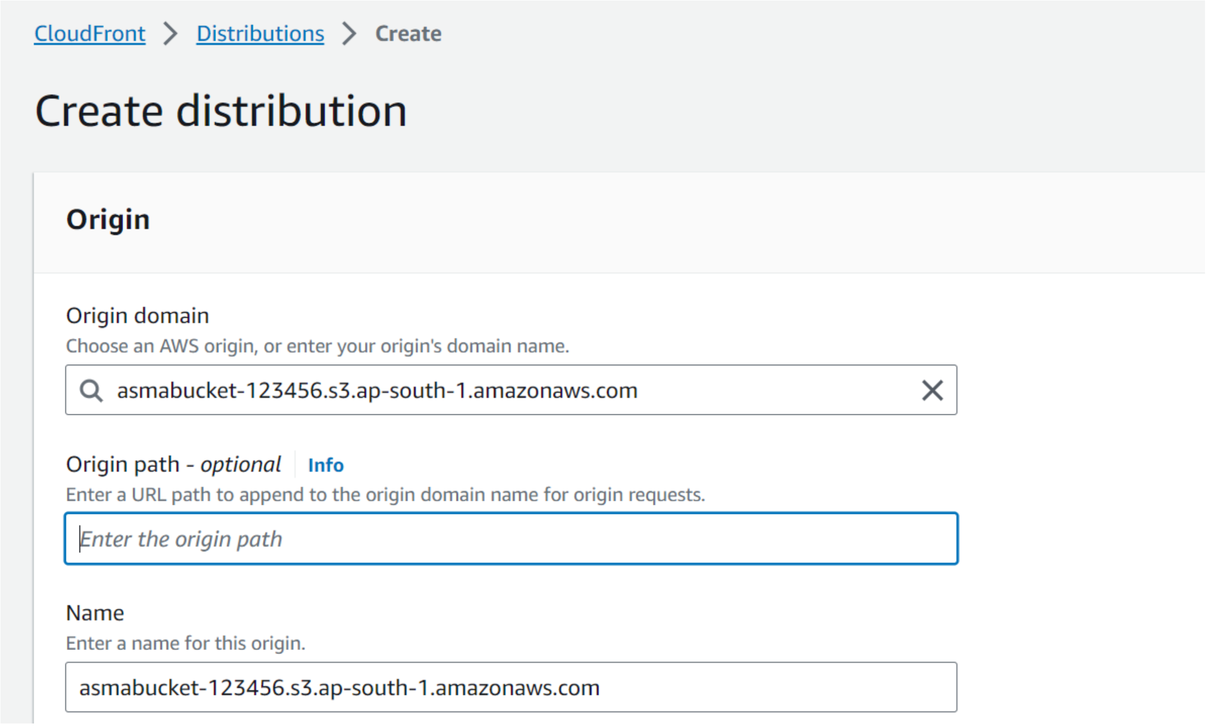
Task 4: Creating a CloudFront Distribution

1. Navigate to **CloudFront** by clicking on the **Services** menu at the top, then click on **CloudFront** in the **Network and Content Delivery** section.
2. Click on the **Create a CloudFront distribution** button.

3. Now Configure distribution as follows:

- **Origin Domain Name**: On click of input space, Select your S3 bucket:

https://asmabucket-123456.s3.ap-south-1.amazonaws.com



CloudFront > Distributions > Create

Create distribution

Origin

Origin domain

Choose an AWS origin, or enter your origin's domain name.

asmabucket-123456.s3.ap-south-1.amazonaws.com

Origin path - *optional* Info

Enter a URL path to append to the origin domain name for origin requests.

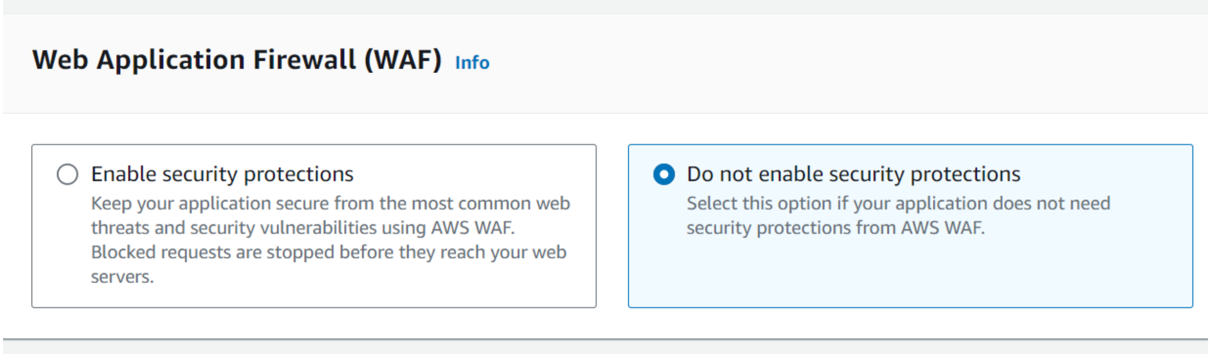
Enter the origin path

Name

Enter a name for this origin.

asmabucket-123456.s3.ap-south-1.amazonaws.com

4. Choose **Do not enable security protections** under **Web Application Firewall(WAF)**.



Web Application Firewall (WAF) Info

○ Enable security protections
Keep your application secure from the most common web threats and security vulnerabilities using AWS WAF. Blocked requests are stopped before they reach your web servers.

● Do not enable security protections
Select this option if your application does not need security protections from AWS WAF.

5. Leave everything as default, scroll down, and click on the **Create distribution** button.

6. You can see that the CloudFront distribution is **enabled** successfully.
Note: This process will take around 5-10 minutes.

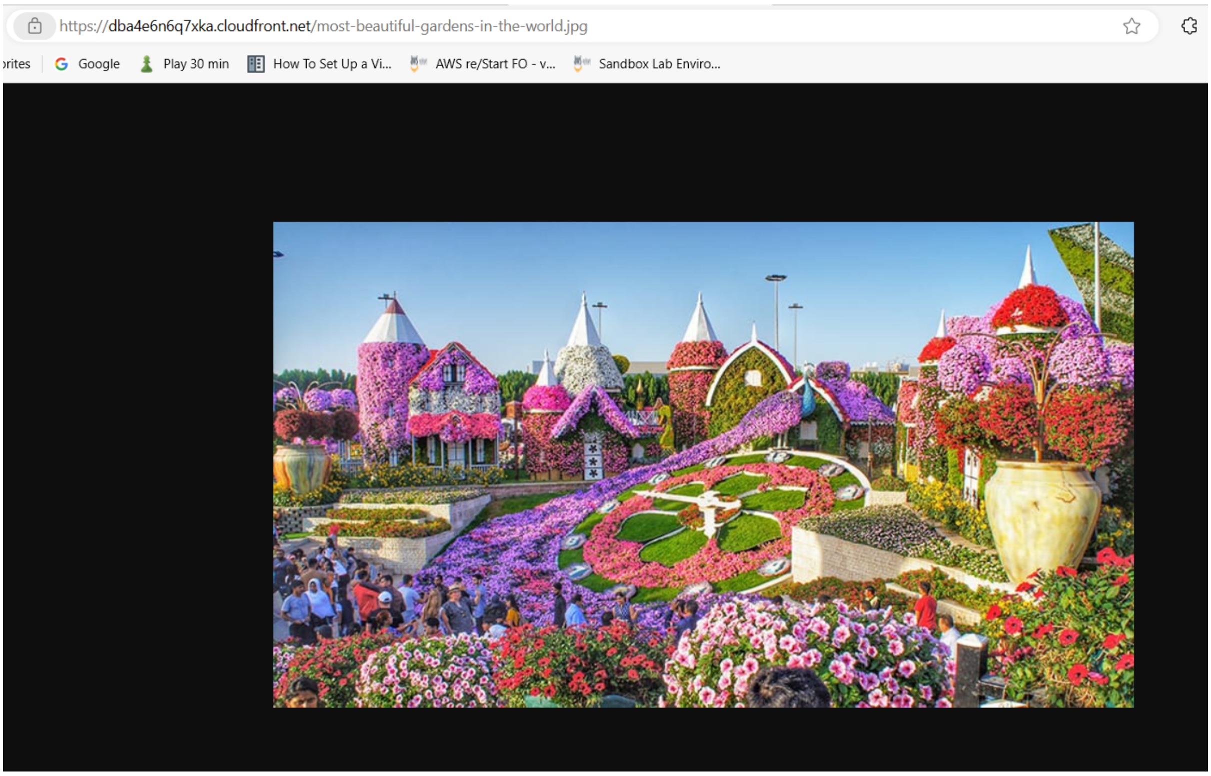
7. The domain name that Amazon CloudFront assigns to your distribution appears in the list of distributions.

https://dba4e6n6q7xka.cloudfront.net

Task 5: Accessing Image through CloudFront

Amazon CloudFront is now pointed to Amazon S3 bucket origin and you know that the domain name is associated with the distribution. You can create a link to the image in the Amazon S3 bucket with that domain name.

1. For testing your distribution, copy your domain name and append your image name after the domain name.
    - **Example:** https://dba4e6n6q7xka.cloudfront.net**/most-beautiful-gardens-in-the-world.jpg**
2. Open the CloudFront URL in a new tab. You can see your uploaded image.

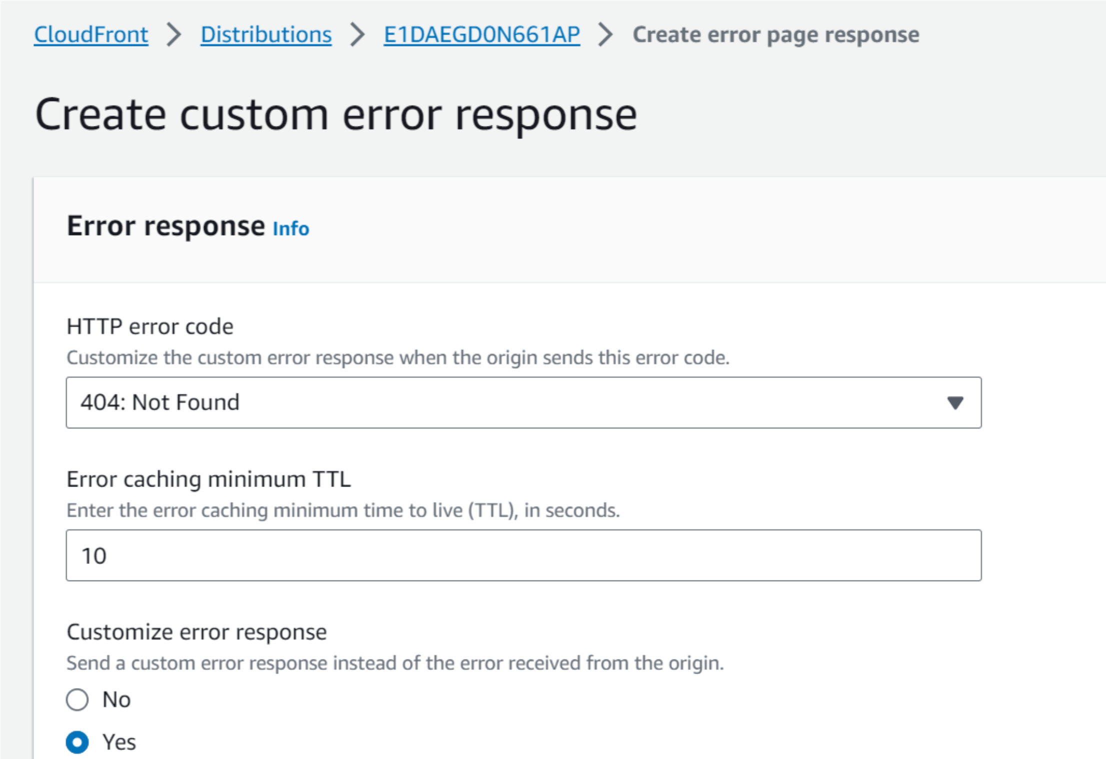


3. You can see how much faster the CloudFront URL image loads as compared to the S3 URL. When end users request an object using a CloudFront domain name, they are automatically routed to the nearest edge location for high-performance delivery of your content.

Task 6 : Configuring Custom Error Page

1. Navigate back to **CloudFront Dashboard** and select the **distribution** created.
2. Select the **Error pages** tab.
    1. Click on the **Create custom error response** button.
    2. Now we need to set up our custom error page:
        - **HTTP Error Code**: Select **404: Not Found**
        - **Error Caching Minimum TTL**: Enter ***10***
        - **Customize Error Response**: Select **Yes**
        - **Response Page Path**: Enter ***/CustomErrors/error.html***
        - **HTTP Response Code**: Select **404: Not Found**

- Click on **Create custom error response** button.



CloudFront > Distributions > E1DAEGD0N661AP > Create error page response

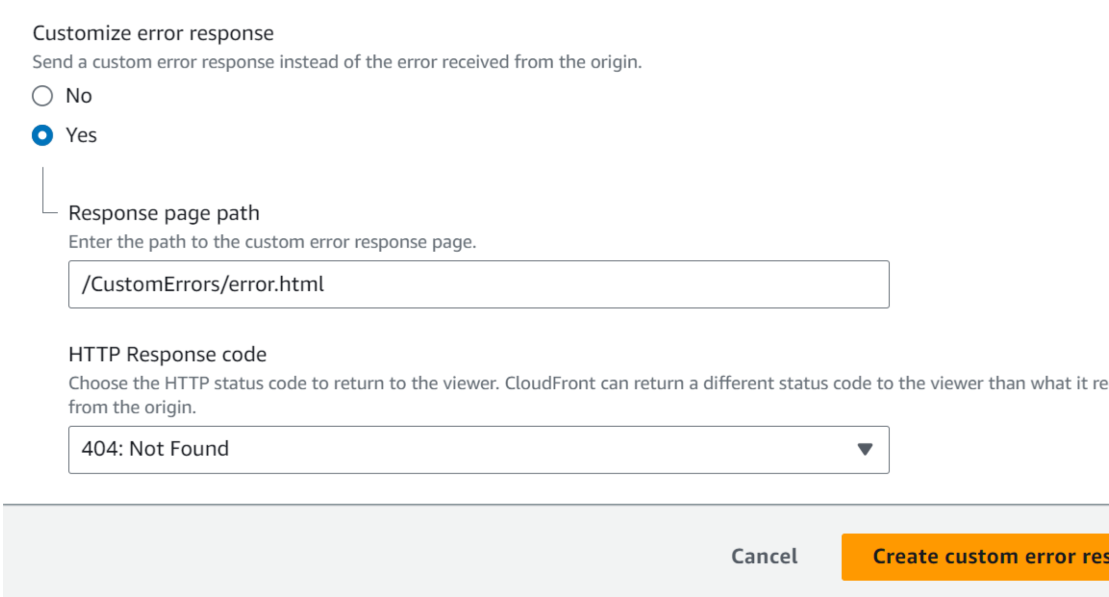
Create custom error response

Error response Info

HTTP error code
Customize the custom error response when the origin sends this error code.
404: Not Found

Error caching minimum TTL
Enter the error caching minimum time to live (TTL), in seconds.
10

Customize error response
Send a custom error response instead of the error received from the origin.
○ No
● Yes



Customize error response
Send a custom error response instead of the error received from the origin.
○ No
● Yes

Response page path
Enter the path to the custom error response page.
/CustomErrors/error.html

HTTP Response code
Choose the HTTP status code to return to the viewer. CloudFront can return a different status code to the viewer than what it re
from the origin.
404: Not Found

3. Navigate back to **Distributions** and wait for your distribution to complete state to change **Deploy.**
    1. **Note**: This process will take around 5-10 minutes.
    2. Once the state has been changed to **Deploy**, we will test the error page.
    3. Enter the URL of an image that does not exist in your S3 bucket with the CloudFront domain name.
       eg: **https://dba4e6n6q7xka.cloudfront.net/abcimage.png**

4. If you can see your HTML error page in the browser, it means you successfully set up your custom error page.

   Task 7 : Restricting the Geographic Distribution of Your Content

   If you need to prevent users in selected countries from accessing your content, you can specify either a whitelist (countries where they can access your content) or a blacklist (countries where they cannot) by using restrictions.

   1. On the distribution settings page, select **Geographic locations tab** and click on **Edit** button.
      - **Restriction Type:** Select **Block list**
      - **Select the country where you are currently** and click on it to check this option.
      - On enabling this option, the request from the specified country which is "Blacklisted", will not be displayed and a default error message is displayed.
      - Click on **Save changes** button

        Go to the distribution list and wait for your distribution to complete the state changed to **deployed**.

        7. Once the state has been changed to **deployed**, we will test the restriction through CloudFront in the browser.
        8. You can see the following error message:
           - **403: Error The Amazon CloudFront distribution is configured to block access from your country.??**
      - 3. Let us configure a custom error page:
        4. Navigate back to **CloudFront Dashboard** and select the **distribution** you have created.
        5. On the settings page, select **Error pages** tab.
        6. Click on the **Create custom error response** button.
        7. Now we need to set up our custom error page:
           7. **Http Error Code**: Select **403: Forbidden**
           8. **Error Caching Minimum TTL**: Enter ***10***
           9. **Customize Error Response**: Select **Yes**

10. **Response Page Path**: Enter ***/CustomErrors/block.html***
11. **HTTP Response Code**: Select **403: Forbidden**
12. Click on **Create custom error response** button.

8. Navigate back to **Distributions** and wait for your distribution to complete state to change **Deploy.**
9. **Note:** This process will take around 5-10 minutes.
10. Once the state has been changed to **Deploy**, we will test the restriction through CloudFront in the browser.
    7. If you see the error, this means you successfully configured a custom error page and restricted image access from your country.