**Lab Steps**

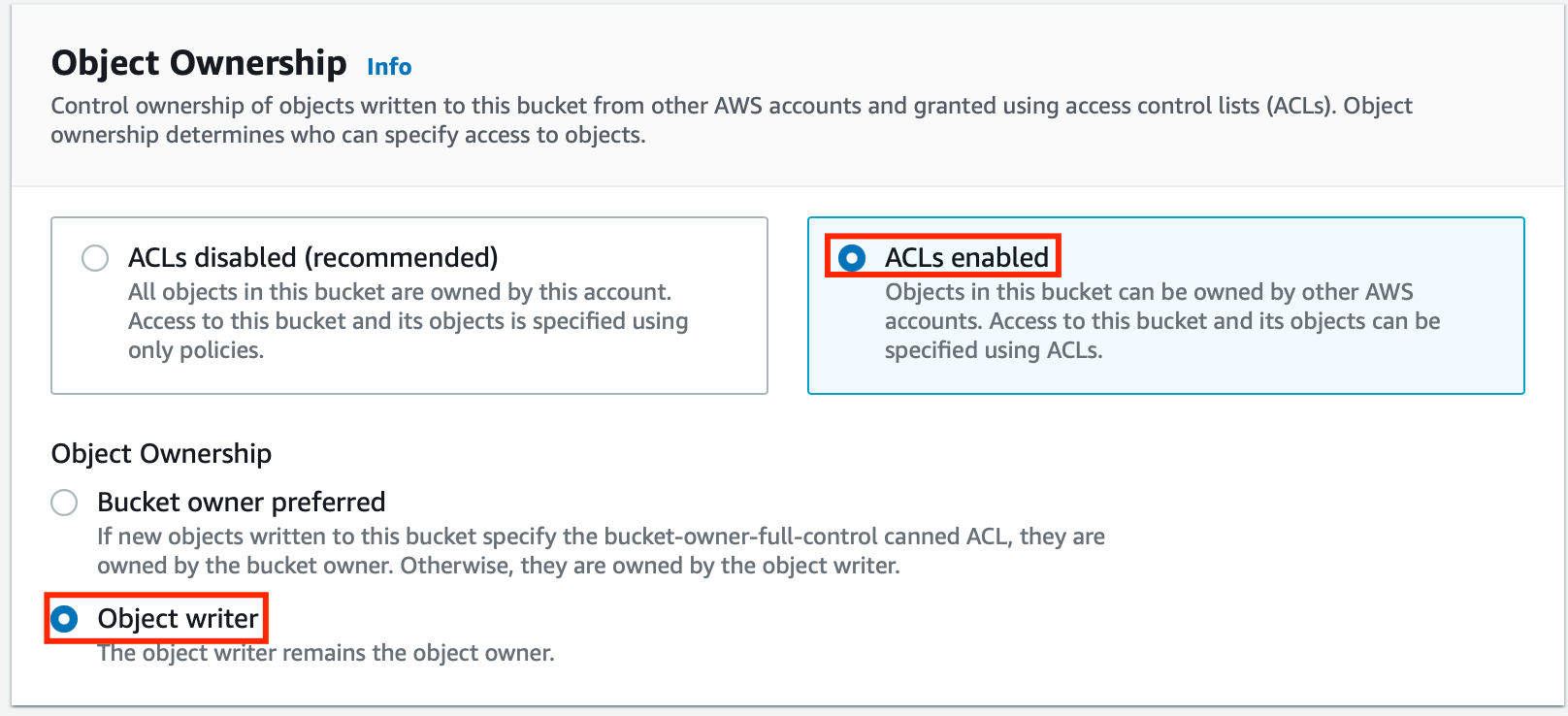
Task 1: Sign in to AWS Management Console

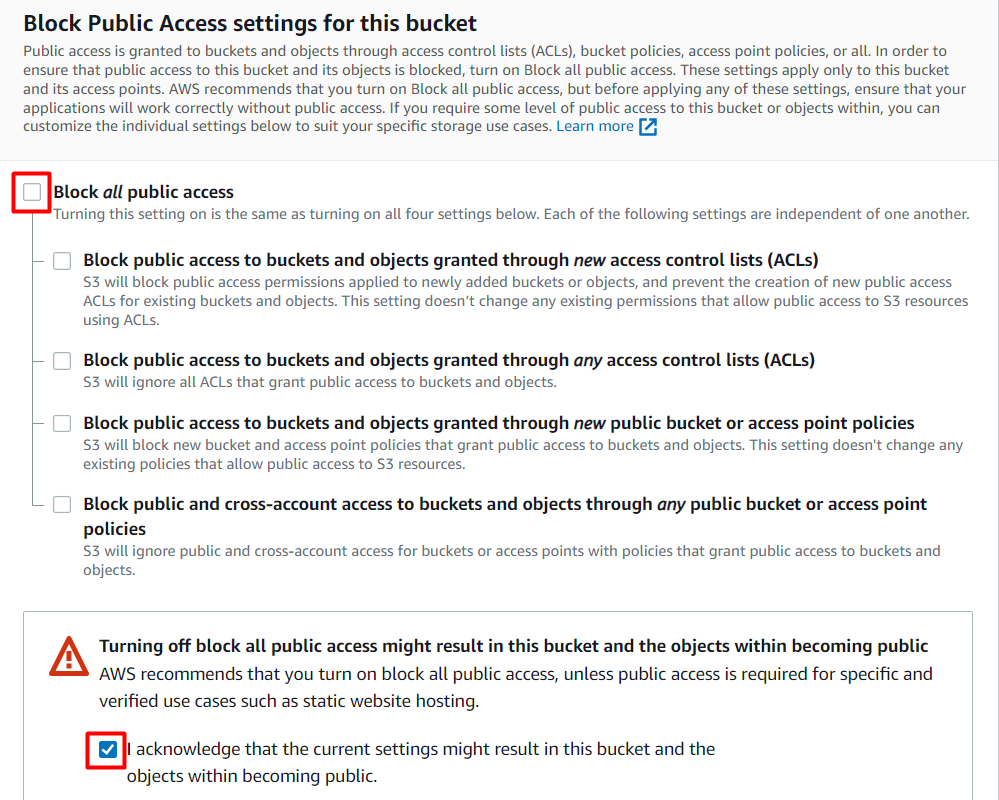
1. Click on the  button, and you will get redirected to AWS Console in a new browser tab.
2. On the AWS sign-in page,
   * Leave the Account ID as default. Never edit/remove the 12 digit Account ID present in the AWS Console. otherwise, you cannot proceed with the lab.
   * Now copy your **User Name** and **Password** in the Lab Console to the **IAM 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 **US East (N. Virginia) us-east-1.**

Task 2: Create 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 enabling ACLs, and allowing public access.

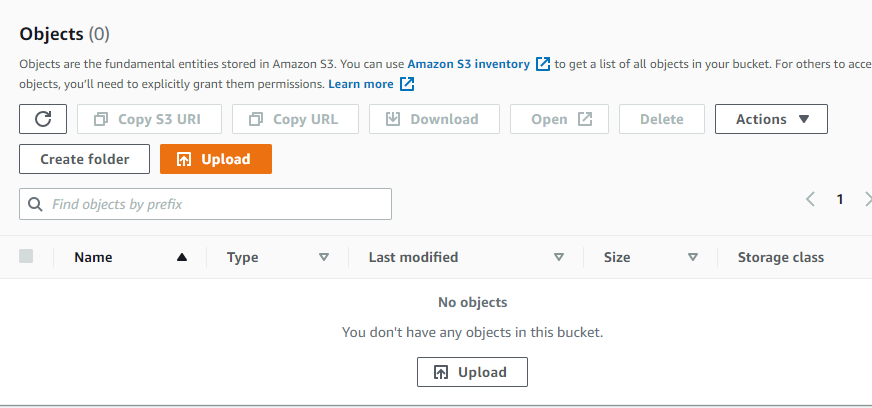
1. Make sure you are in the **US East (N. Virginia) us-east-1** Region.
2. Navigate to the  menu at the top. Click on **S3** in the **Storage** section.
3. In the S3 dashboard, click on the  and fill in the bucket details.
   * Bucket name:Enter ***whizlabs1234567***
     + **Note: S3 Bucket names are globally unique, choose a name that is available.**
   * AWS Region:Select **US East (N. Virginia) us-east-1**
   * Object Ownership: Select **ACLs enabled** option and choose **Object writer** as Object owner



* + Scroll down to **Block Public Access settings for this bucket** and **Uncheck** the **Block all Public Access** and **acknowledge** the change.  
    
  + No need to change anything further, just click on the .

Task 3: Upload a file to an S3 bucket

1. Click on the bucket name you just created and you can see that there are no objects created in the bucket.

****

1. You can upload any image from your local machine or you can download our test image from [**Download Me**](https://labresources.whizlabs.com/35b5214b5298cb78a0d4e287e6698091/whizlabs_logo_49_31.png).
2. To upload a file to our S3 bucket,
   * Click on the **Upload** button.
   * Click on **Add files**.
   * Browse for your local image or the image we provided and select it.
   * Click on the **Upload** button.
   * You can watch the progress of the upload from within the transfer panel at the bottom of the screen.
   * Once your file has been uploaded, click on **Close** and you can see an object in the bucket.

Graphical user interface, text, application, chat or text message

Description automatically generated

Task 4: Creating Custom Error Pages

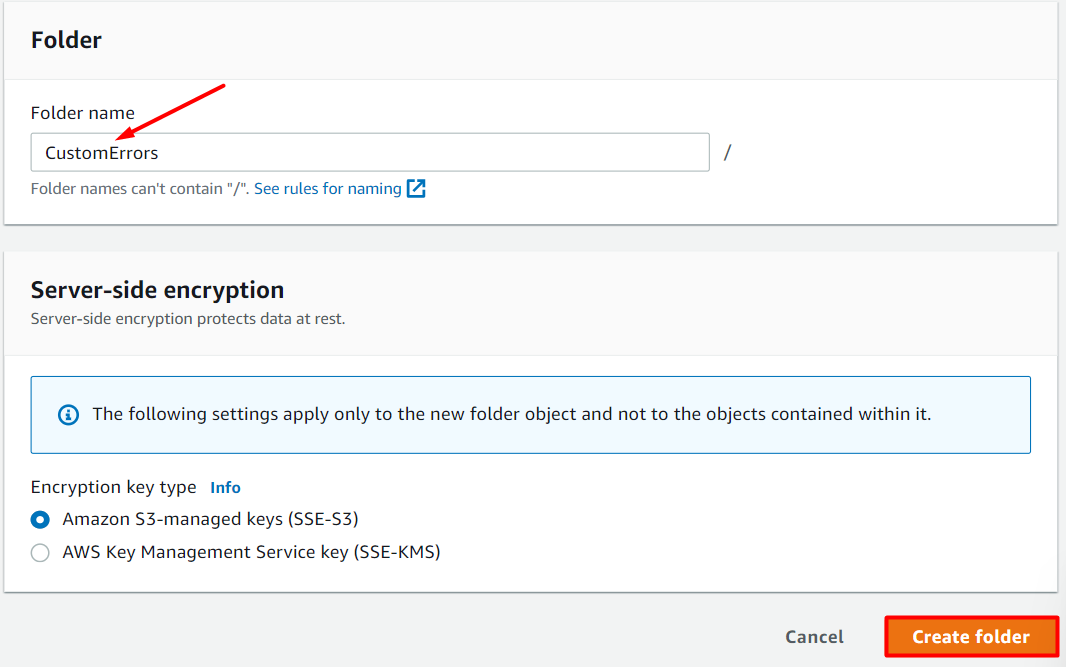
In this task, we will learn how to create customized error pages for CloudFront. These pages will be displayed in the event that an origin returns an HTTP 4xx or 5xx error. To do this, we must ensure that the error pages are stored in a location that CloudFront can access. In this case, we will use the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

1. To set up a custom error page, access the S3 bucket by clicking on it.
2. Click on **Create Folder** button and create a folder with the name ***CustomErrors***

Graphical user interface, text, application, email

Description automatically generated

1. Select **Server-side encryption** as **Amazon S3-managed keys (SSE-S3).**
2. Click on the **Create folder** button.



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:

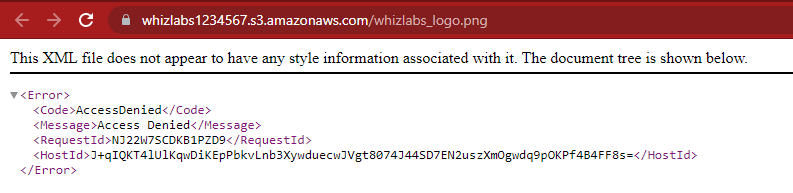
<html><h1>This is Error Page</h1></html>

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>

1. Use the**Upload** button to upload the **block.html** file in the folder.

Task 5: 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://whizlabs1234567.s3.amazonaws.com/whizlabs\_logo.png**
   * You will see the **AccessDenied** message, meaning the object is not publicly accessible.
4. Go back to the Bucket and click the **Permissions** tab.
5. Scroll down to the **Bucket Policy** and click on **Edit**. **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>/\*"          }      ]  } |

1. Open the Image **Object URL** again or refresh the one already open.
2. If you can see your uploaded image in the browser, it means your image is publicly accessible. If not, check your bucket policy again.

Graphical user interface

Description automatically generated with medium confidence

Task 6: 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. Navigate to the left side and click on icon. Click on **Distributions.**
3. Now click on the 
4. Now Configure distribution as follows:
   * **Origin Domain Name**:
     + On click of input space, Select your S3 bucket: **whizlabs1234567.s3.us-east-1.amazonaws.com**
5. No need to change anything in the configuration, scroll down and click on the 
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. It will look similar to https://d1jptzlydefk0d.cloudfront.net

Task 7: 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.

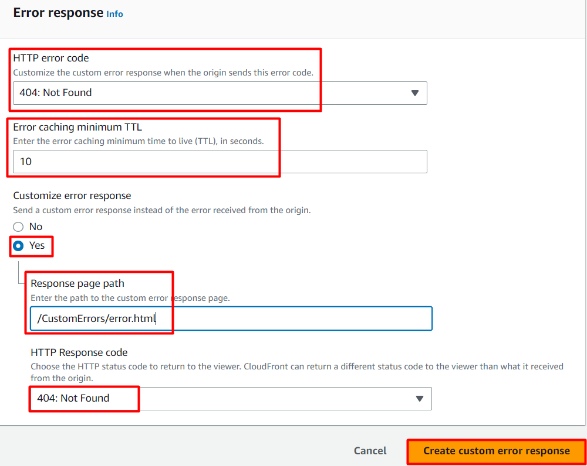
1. For testing your distribution, copy your domain name and append your image name after the domain name.
   * **Example: https://d1jptzlydefk0d.cloudfront.net/whizlabs\_logo.png**
2. Open the CloudFront URL in a new tab. You can see your uploaded image.
3. You can see how much faster the CloudFront 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.

Graphical user interface, text

Description automatically generated

Task 8 : Configuring Custom Error Page

1. Navigate back to **CloudFront Dashboard** and select the **distribution** created.
2. Select the **Error pages** tab.
   * Click on the 
   * 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 



1. Navigate back to **Distributions** and wait for your distribution to complete state to change **Deploy.**
   * **Note**: This process will take around 5-10 minutes.
   * Once the state has been changed to **Deploy**, we will test the error page.
     + Enter the URL of an image that does not exist in your S3 bucket with the CloudFront domain name
       - https://d1gp7t6bst6xdp.cloudfront.net/abc.png

Text

Description automatically generated

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

Task 9 : 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 
   * **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.

Graphical user interface, text, application, email

Description automatically generated

1. Go to the distribution list and wait for your distribution to complete the state changed to **deployed**.
   * Once the state has been changed to **deployed**, we will test the restriction through CloudFront in the browser.
     + ***https://d1jptzlydefk0d.cloudfront.net/whizlabs\_logo.png***
   * You can see the following error message:
     + **403: Error The Amazon CloudFront distribution is configured to block access from your country.  
       Graphical user interface, application, Word

       Description automatically generated**
2. Let us configure a custom error page:

* Navigate back to **CloudFront Dashboard**and select the **distribution**you have created.
* On the settings page, select **Error pages** tab.
  + Click on the 
  + Now we need to set up our custom error page:
    - **Http Error Code**:Select **403: Forbidden**
    - **Error Caching Minimum TTL**: Enter ***10***
    - **Customize Error Response**: Select **Yes**
    - **Response Page Path**: Enter ***/CustomErrors/block.html***
    - **HTTP Response Code**: Select **403: Forbidden**
    - Click on 

1. Navigate back to **Distributions** and wait for your distribution to complete state to change **Deploy.**
2. **Note:** This process will take around 5-10 minutes.
3. Once the state has been changed to **Deploy**, we will test restriction through CloudFront in the browser.
   * ***https://d1jptzlydefk0d.cloudfront.net/whizlabs\_logo.png***

Text

Description automatically generated

1. If you see the error, this means you successfully configured a custom error page and restricted image access from your country.

Task 10 : Validation Test

1. Once the lab steps are completed, please click on the  on the left side panel.
2. This will validate the resources in the AWS account and displays whether you have completed this lab successfully or not.
3. Sample output :

Graphical user interface, text, application, email

Description automatically generated

**Completion and Conclusion**

1. You have successfully created an Amazon CloudFront distribution and published an image through CloudFront.
2. You learned how to configure Custom Error Pages for CloudFront Distribution.
3. You learned how to configure restrictions based on Geo-location.
4. You have successfully validated the lab.

**End Lab**