

Introduction to Amazon CloudFront

Level: **Intermediate**

Amazon S3 Amazon CloudFront Amazon Web Services

English ▼



Your last attempt on **04-Jan-2025**

[View all](#)

Average Start time

Less than a minute

[Start Guided Lab →](#)

Lab Overview

Lab Steps



Cloud Architect, Cloud Network Engineer



Storage, Networking

Lab Steps

Task 1: Sign in to AWS Management Console

1. Click on the **Open Console** 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 **Services** menu at the top. Click on **S3** in the **Storage** section.
3. In the S3 dashboard, click on the **Create bucket** 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

Object Ownership [Info](#)
Control ownership of objects written to this bucket from other AWS accounts and granted using 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.

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.

- Scroll down to **Block Public Access settings for this bucket** and **Uncheck** the **Block all Public Access** and **acknowledge** the change.

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.



Block public access to buckets and objects granted through *new* public bucket or access point policies

S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.



Block public and cross-account access to buckets and objects through *any* public bucket or access point policies

S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.



Turning off block all public access might result in this bucket and the objects within becoming public

AWS recommends that you turn on block all public access, unless public access is required for specific and verified use cases such as static website hosting.



I acknowledge that the current settings might result in this bucket and the objects within becoming public.

- No need to change anything further, just click on the **Create bucket** button.

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.

Objects (0)

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 objects, you'll need to explicitly grant them permissions. [Learn more](#)



Name	Type	Last modified	Size	Storage class
------	------	---------------	------	---------------

No objects

You don't have any objects in this bucket.

2. You can upload any image from your local machine or you can download our test image from [Download me](#)

3. To upload a file to our S3 bucket,

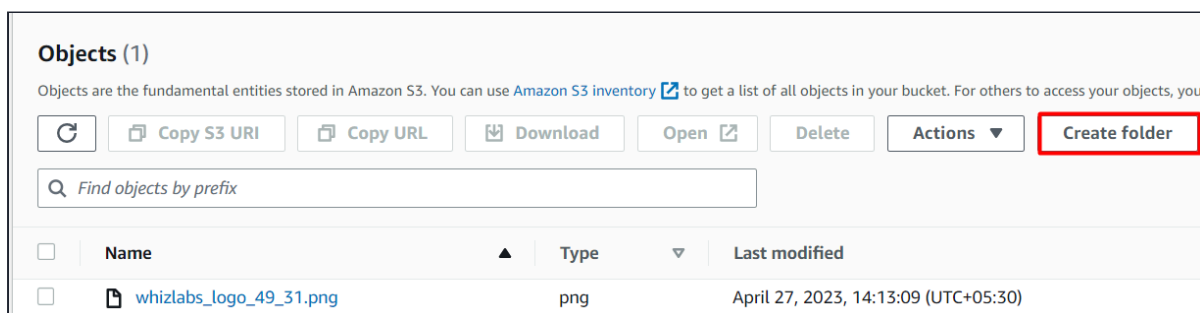
- Click on the **Upload** button.
- Click on **Add files**.
- Browse for image we provided and select it. **{NOTE : use the same image provided to ignore the validation failure}**
- Click on the **Upload** button.
- You can watch the progress of the upload from within the transfer panel at the bottom of the screen.
- Once your file has been uploaded, click on **Close** and you can see an object in the bucket.

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**



3. For **Server-side encryption** Select **Specify an encryption key** keep rest things as default.

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)

4. Click on the **Create folder** button.

5. Click on the new **CustomErrors** folder.

6. 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>
```

7. Use the **Upload** button to upload the **error.html** file in the folder.

8. 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>
```

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

Task 5: Making the objects public

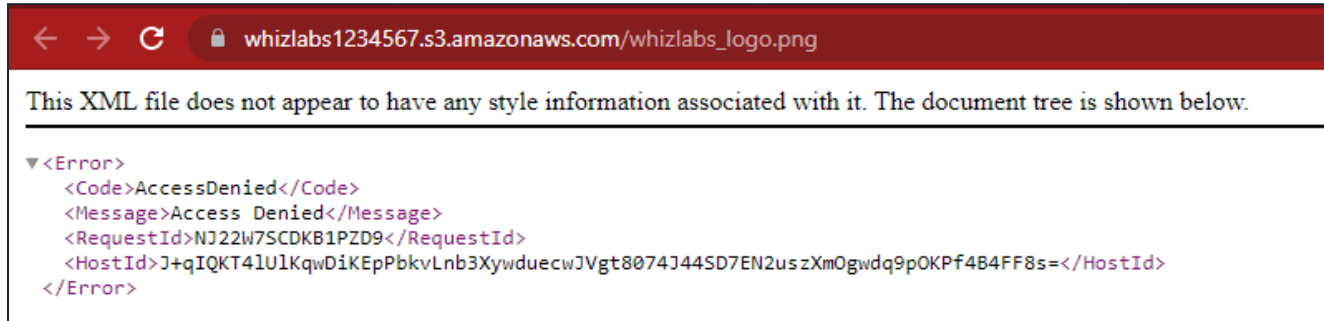
1. Click on the image name. You can see the image details like Owner, size, link, etc.

2. Copy the Object URL and paste it into a new tab.

3. A sample **Object URL**:

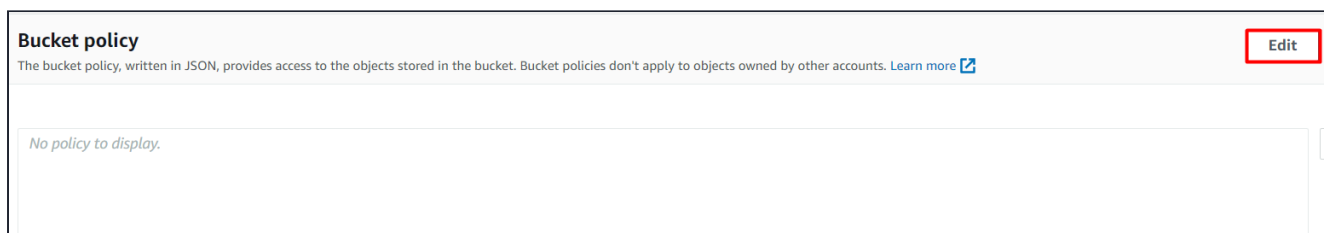
`https://whizlabs1234567.s3.amazonaws.com/whizlabs_logo_58_32.png`

- You will see the **AccessDenied** message, meaning the object is not publicly accessible.**`whizlabs_logo_58_32.png`**



4. Go back to the Bucket and click the **Permissions** tab.

5. Scroll down to the **Bucket Policy** and click on **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>/*"
    }
  ]
}
```

Bucket policy

The bucket policy, written in JSON, provides access to the objects stored in the bucket. Bucket policies don't apply to objects owned by other accounts. [Learn more](#)

Bucket ARN

arn:aws:s3:::whizlabs123456722

Policy

1

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.

whizlabs1234567.s3.amazonaws.com/whizlabs_logo.png



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. Click on **Create a CloudFront distribution** button.

Networking & Content Delivery

Amazon CloudFront

Securely deliver content with low latency and high transfer speeds

Amazon CloudFront is a fast content delivery network (CDN) service that securely delivers data, videos, applications, and APIs to customers globally with low latency and high transfer speeds.

Get started with CloudFront

Enable accelerated, reliable and secure content delivery for Amazon S3 buckets, Application Load Balancers, Amazon API Gateway APIs, and more in 5 minutes or less.

[Create a CloudFront distribution](#)

3. Now Configure distribution as follows:

- **Origin Domain Name:** On click of input space, Select your S3 bucket: **whizlabs1234567.s3.us-east-1.amazonaws.com**

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

Web Application Firewall (WAF)

☐ 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.

Task 7: Accessing Image through CloudFront

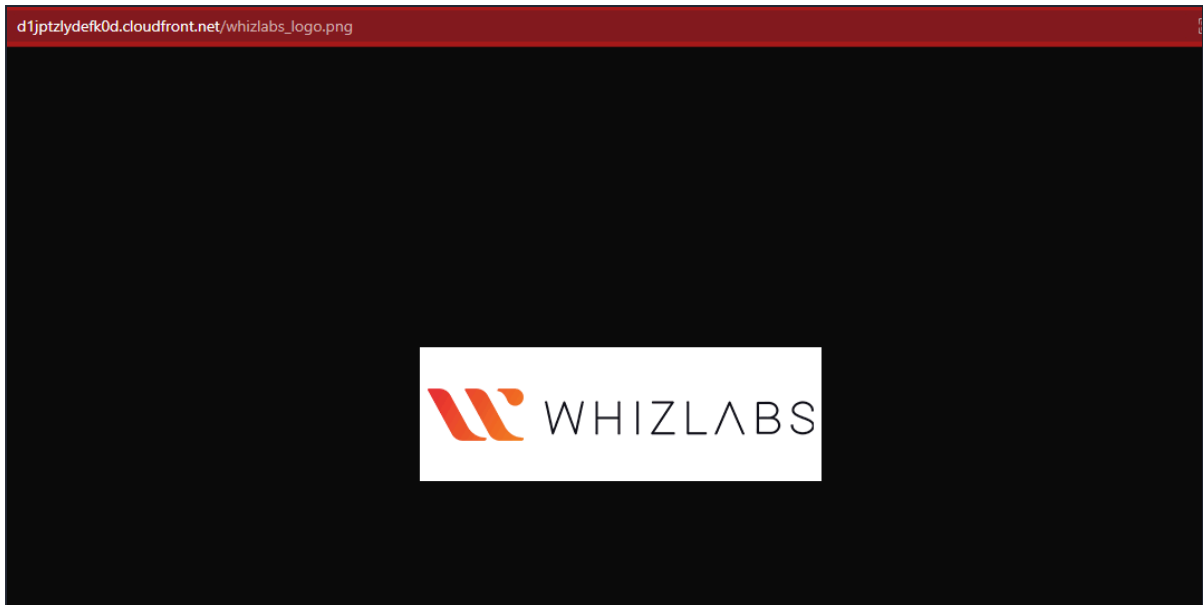
Amazon CloudFront is now pointed to Amazon S3 bucket origin and you know that the domain name is associated with the distribution. You can create a link to the image in the Amazon S3 bucket with that domain name.

1. For testing your distribution, copy your domain name and append your image name after the domain name.

- **Example:** https://dljptzlydefk0d.cloudfront.net/whizlabs_logo_58_32.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.



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

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

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 received from the origin.

404: Not Found

Cancel

Create custom error response

3. 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.

4. 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

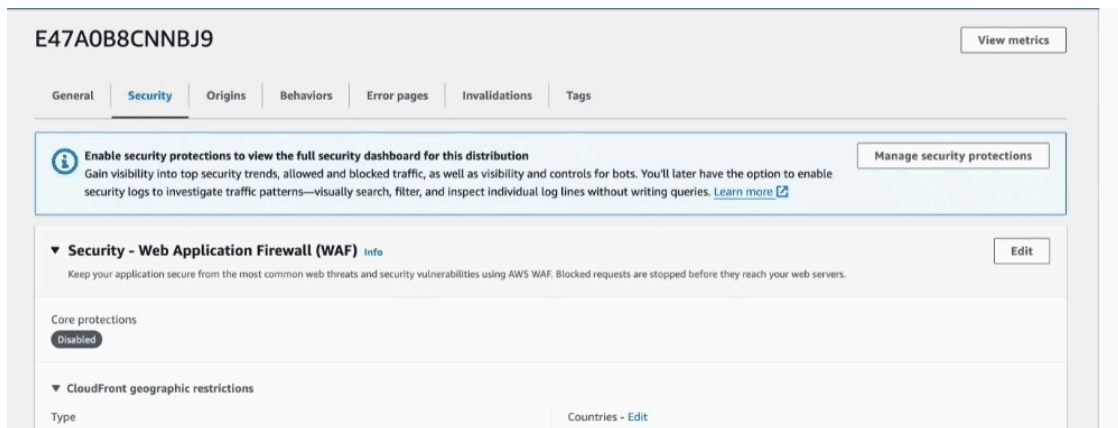
<https://business.whizlabs.com/labs/introduction-to-amazon-cloudfront>

11/15

(countries where they cannot) by using restrictions.

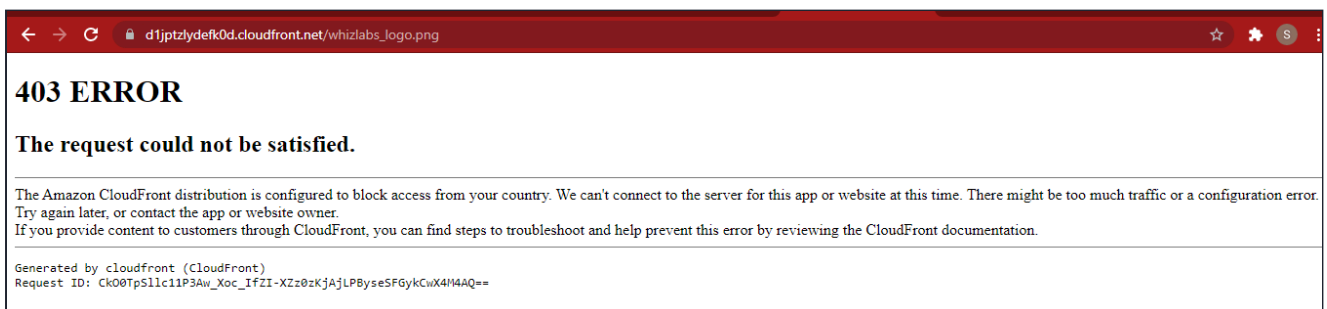
1. On the distribution settings page, select **Security** tab and expand **CloudFront geographic restrictions** click on **Edit** link near Countries.

- **Restriction Type:** Select **Block list**
- **Select the country where you are currently** and click on it to check this option.
- Click on **Save changes** button.



2. 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_58_32.png***
- 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:

- Navigate back to **CloudFront Dashboard** and select the **distribution** you have created.

- On the settings page, select **Error pages** tab.
 - Click on the **Create custom error response** button.
 - 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 **Create custom error response** button.
4. Navigate back to **Distributions** and wait for your distribution to complete state to change **Deploy**.
5. **Note**: This process will take around 5-10 minutes.
6. Once the state has been changed to **Deploy**, we will test the restriction through CloudFront in the browser.

- https://d1jptzlydefk0d.cloudfront.net/whizlabs_logo_58_32.png



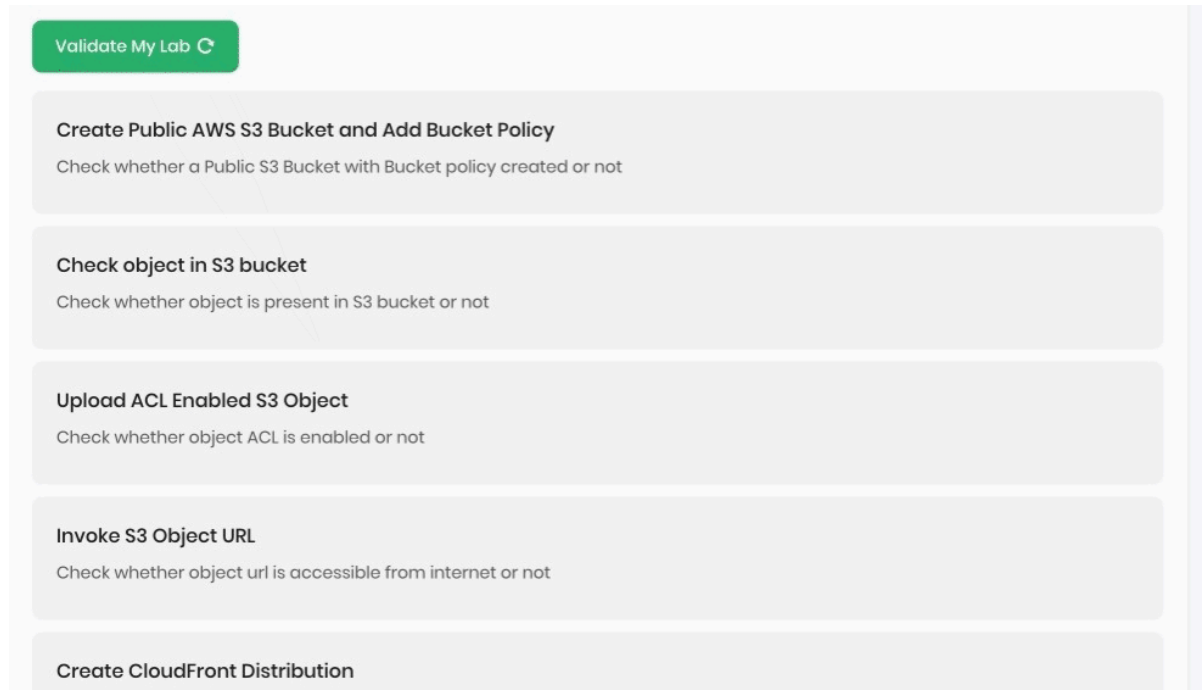
7. If you see the error, this means you successfully configured a custom error page and restricted image access from your country.

Do you know ?

Amazon CloudFront allows you to use custom SSL/TLS certificates, including certificates issued by third-party certificate authorities (CAs), to secure your content delivery. However, what makes this even more interesting is that CloudFront also provides an integrated solution called AWS Certificate Manager (ACM) to simplify the process of managing SSL/TLS certificates.

Task 10: Validation Test

1. Once the lab steps are completed, please click on the **Validation** button 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 :



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

1. Sign out of AWS Account.
2. You have successfully completed the lab.
3. Once you have completed the steps, click on **End Lab** from your whizlabs lab console and wait till the process gets completed.

