Name	Type	Last modified	Size
CustomErrors /	Folder	-	

Project 1: Working with Amazon CloudFront

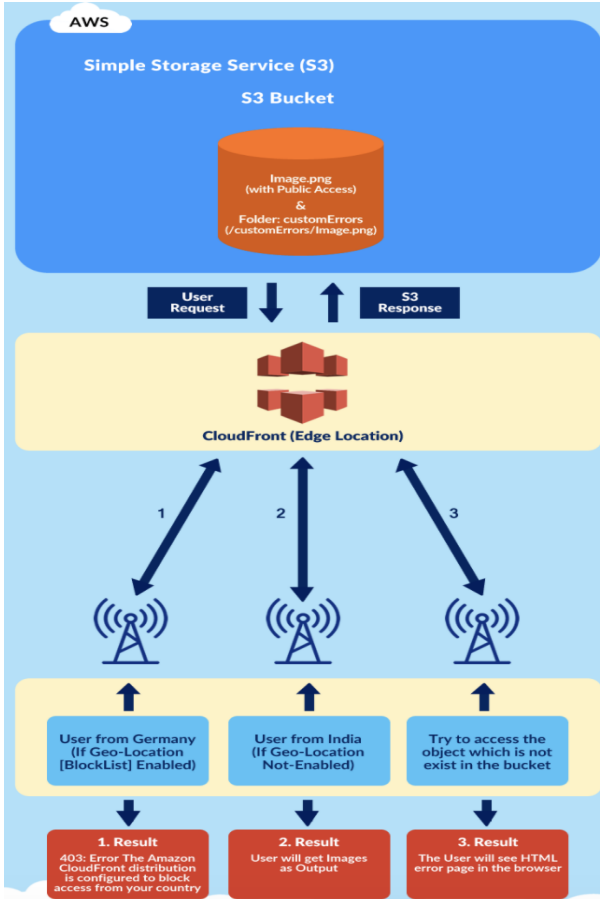
What is Amazon CloudFront?

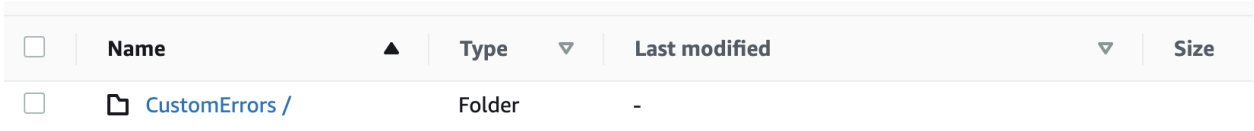
- Amazon CloudFront is a content delivery network offered by AWS. Content Delivery networks provide globally-distributed network of proxy servers which cache content i.e. videos or bulky media, more locally to consumers, thus improving access speed for downloading the content.

Objective of this Project

- We will be demonstrating how a CloudFront Distribution is created. In order for it to work, we will be distributing a publicly accessible image stored in the S3 bucket.

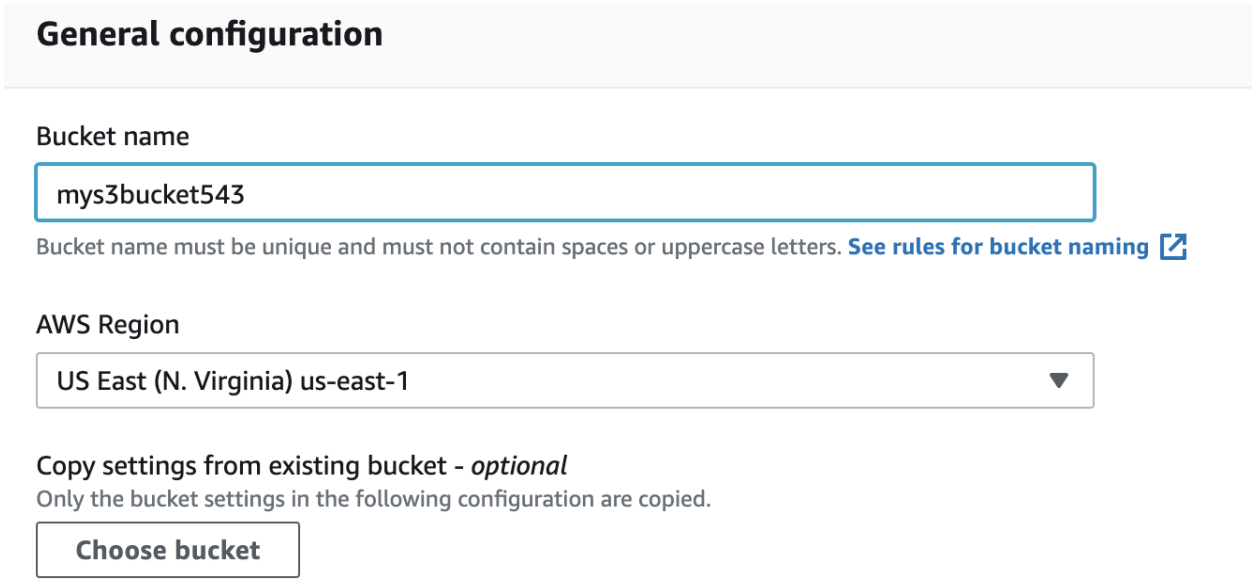
Architecture

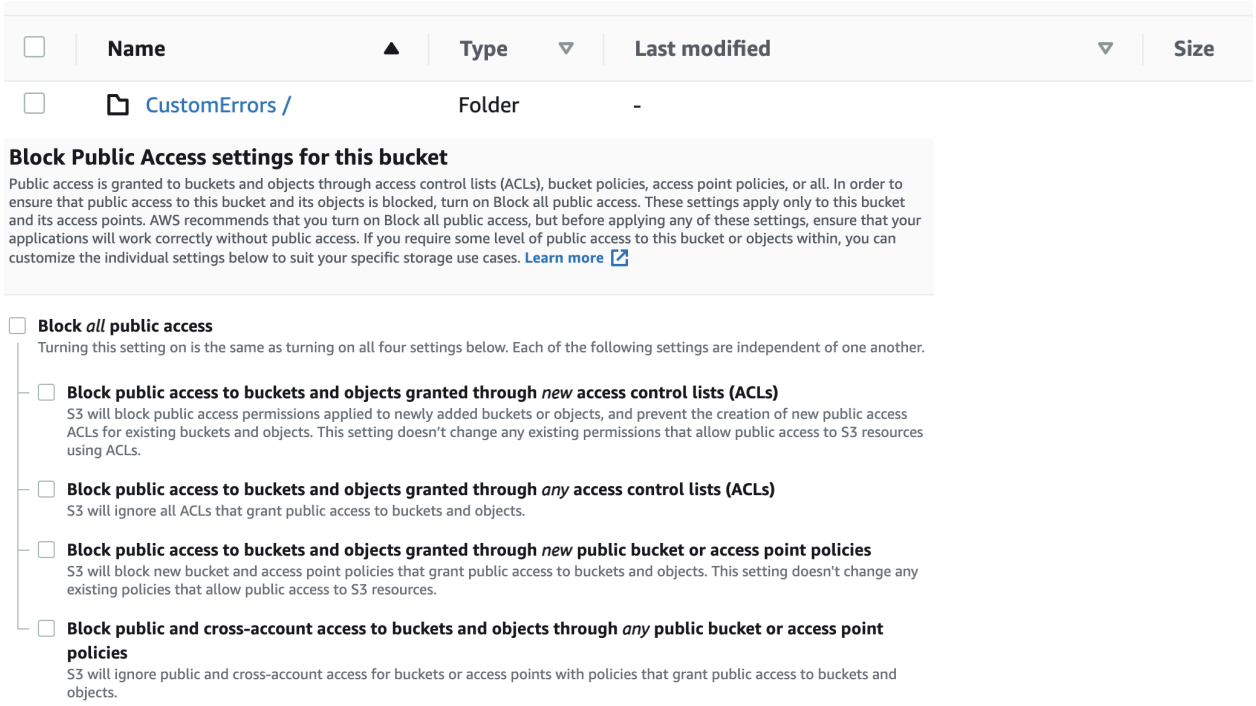




Step 1: Creating an S3 Bucket

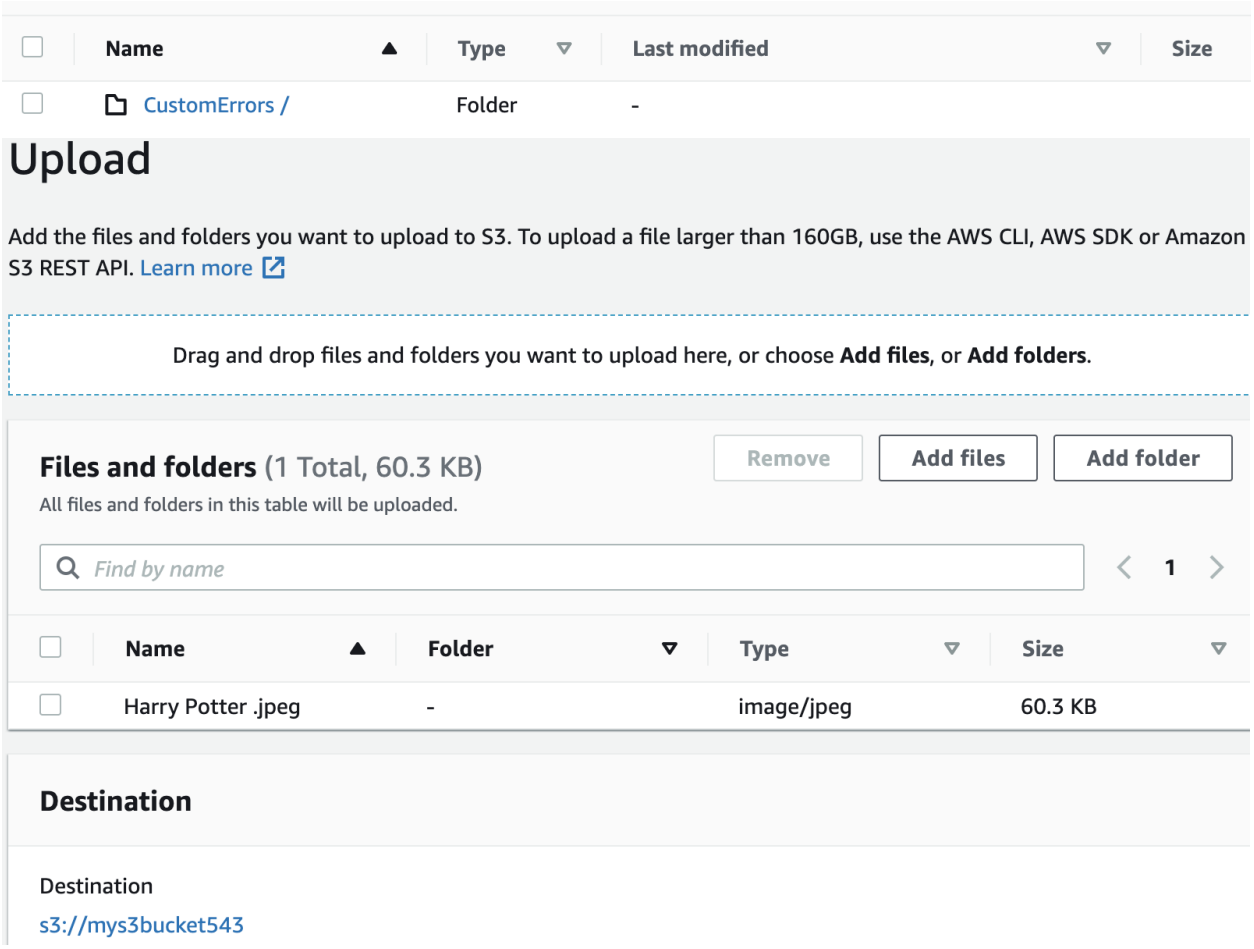
We just created an S3 bucket with the general configuration settings and made sure it was publicly accessible.



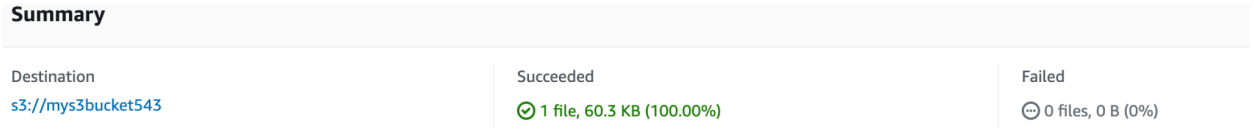


Step 2: Uploading a file to an S3 bucket

We then upload a file into the S3 bucket.



Upload of the file being successful

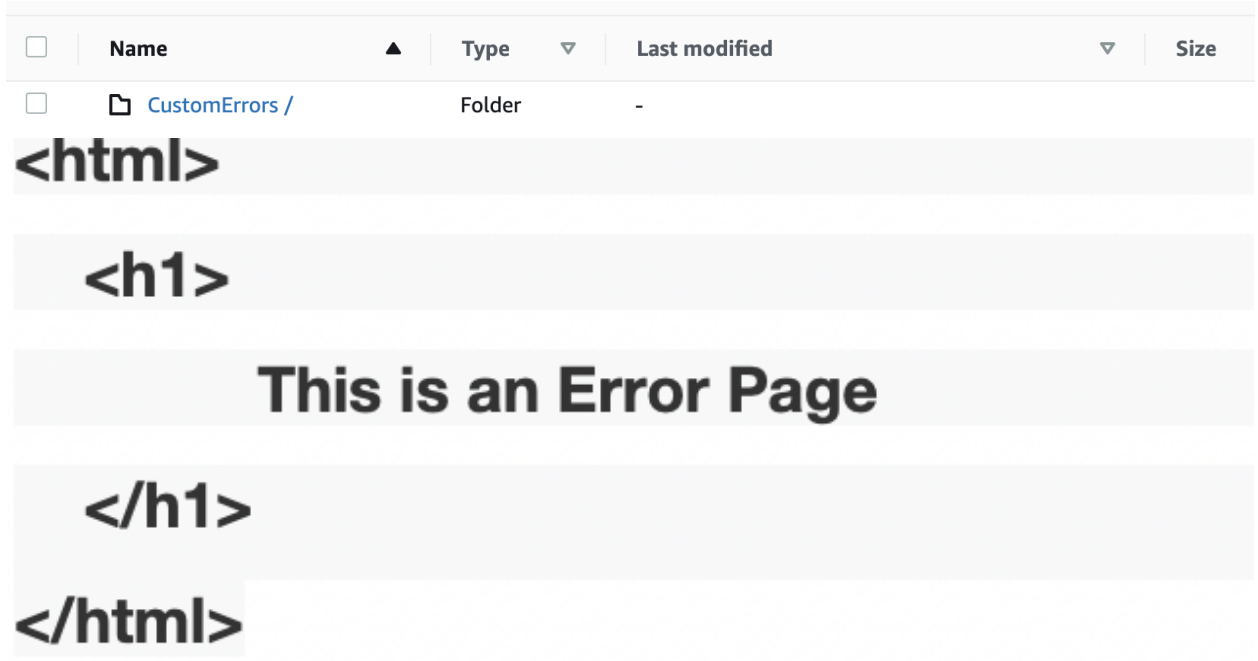


Step 3: Creating Custom Error Pages

The objective of creating custom error pages for CloudFront to return when origin returns HTTP 400 or 500 errors.

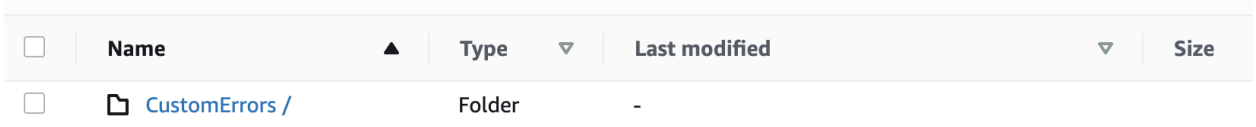
We start with creating a folder, naming it Custom Errors

We then create an error.html file:

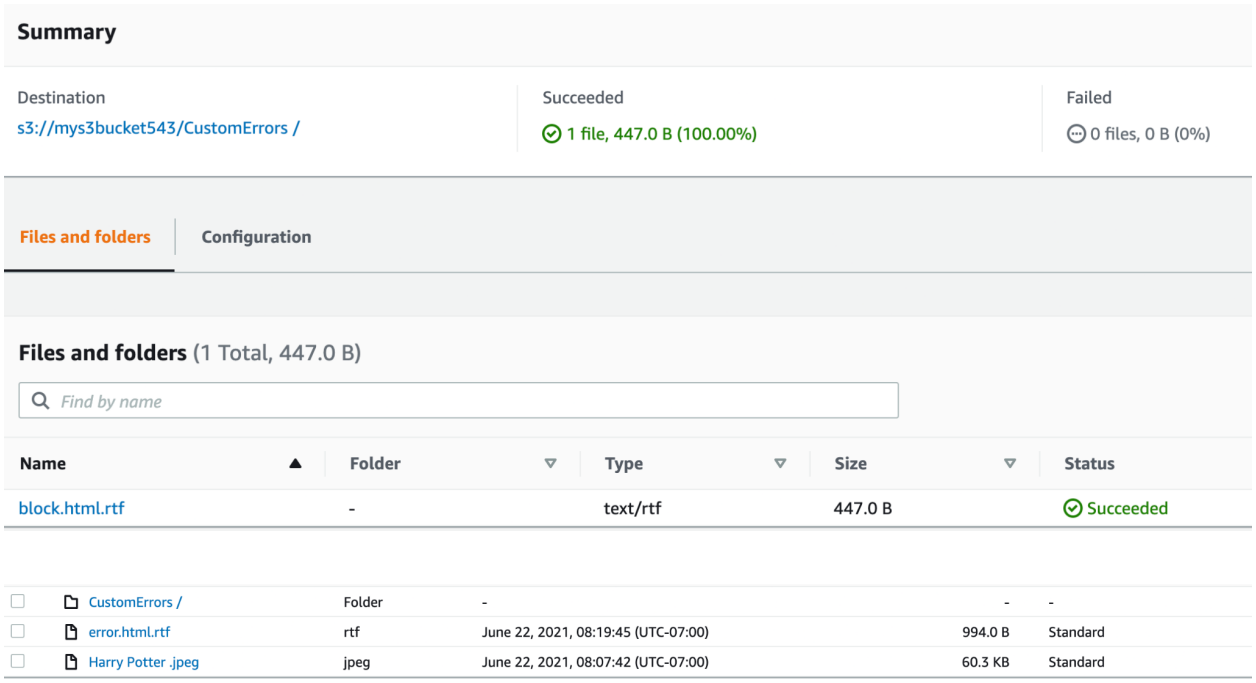


And also a block.html file

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



After creating the two files, we upload error.html on the S3 bucket and upload block.html on the Custom Errors Folder.



Step 4: Making the objects public

When we first click on the image URL, you will see an Access Denied message.

```
<Error>
  <Code>AccessDenied</Code>
  <Message>Access Denied</Message>
  <RequestId>8VJHE1JP6QZYB4P2</RequestId>
  <HostId>9+Ohg/BhfO8CV51RytuLgXScJfwLOqUw0e09tRgp7243HdGRWquoAFaX4u3zEyL2pIx6eT/tYx4=</HostId>
</Error>
```

We immediately update the bucket policy with this, written in JSON:

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Principal": {
            "AWS": "*"
         },
```

Name	Type	Last modified	Size
CustomErrors /	Folder	-	

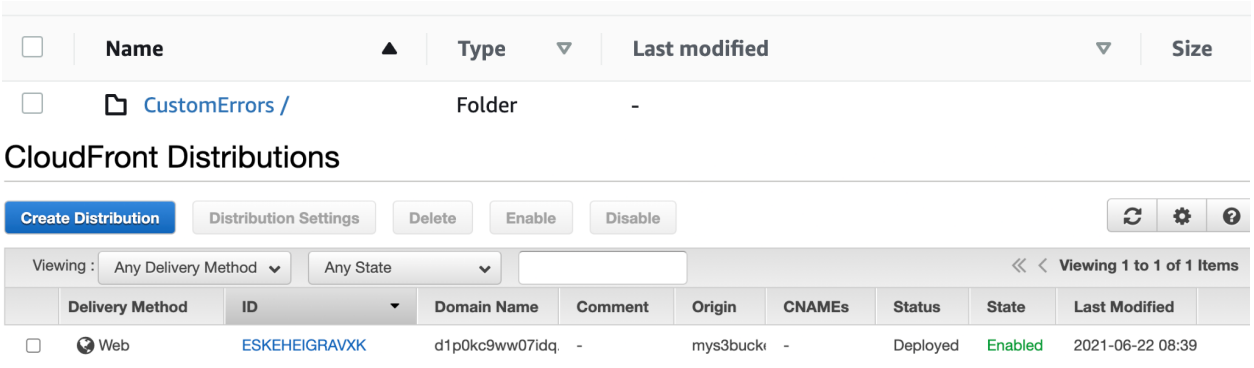
```
        "Action": "s3:ListBucket",
        "Resource": "arn:aws:s3:::mys3bucket543"
    },
    {
        "Effect": "Allow",
        "Principal": {
            "AWS": "*"
        },
        "Action": [
            "s3:GetObject",
            "s3:PutObject"
        ],
        "Resource": "arn:aws:s3:::mys3bucket543/*"
    }
  ]
}
```

Successfully edited bucket policy.

Now that policy has been editing, we click on the URL of the image and the result is this: IT'S WORKING!!!!

Step 5: Creating A CloudFront Distribution

We go on CloudFront and Create A Distribution First where the origin is from the S3 Bucket.



Step 6: Accessing Image Through CloudFront

We then type in the URL found under domain name and attach the file name on the URL.

Domain Name d1p0kc9ww07idq.cloudfront.net

The end result is this. The image loaded faster than what it was supposed to be.

Step 7: Customizing an Error Page

We then create a customized error page using the error.html file that can be displayed if an error on the CloudFront URL were to occur.

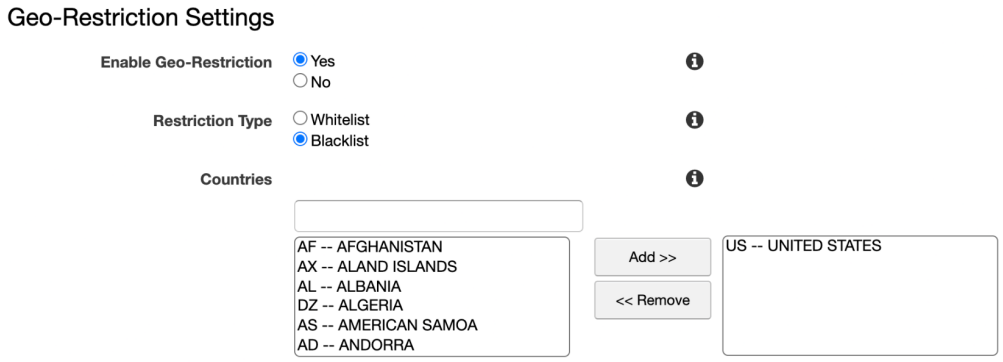
We test it on the browser and see if it works or not by utilizing a random URL page. The end result is this.

This is an Error Page

Step 8: Restricting the Geographic Distribution of Your Content

If we need to prevent users in selected countries from accessing content, we can either whitelist or blacklist content by using restrictions.

Here’s how this process is done.



We will then test the following to see how it works. The end result is a 403 Error.

403 ERROR

The request could not be satisfied.

The Amazon CloudFront distribution is configured to block access from your country. We can't connect to the server for this app or website at this time. '