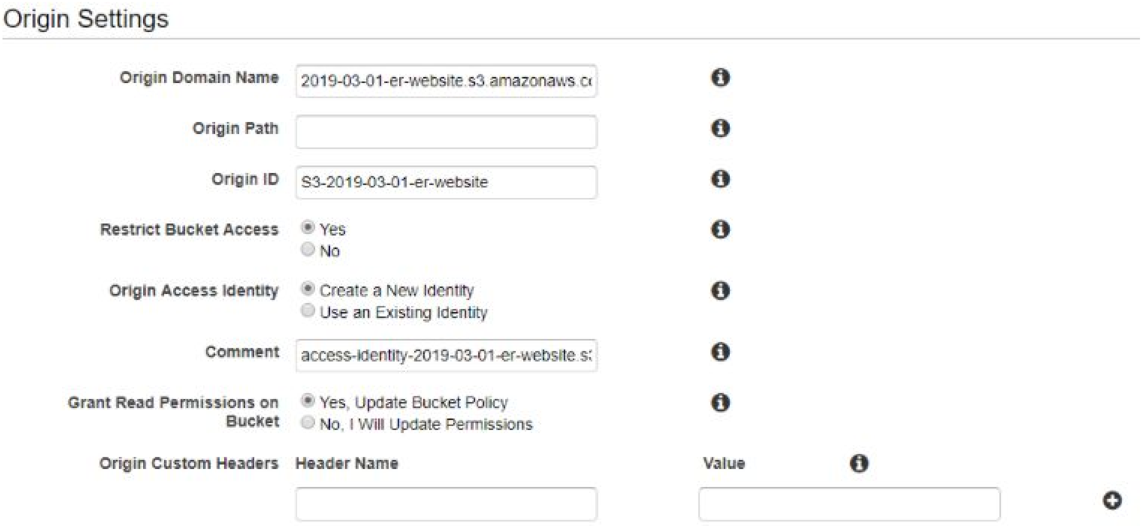
Creating a CloudFront Distribution

1. Steps for creating a CloudFront distribution

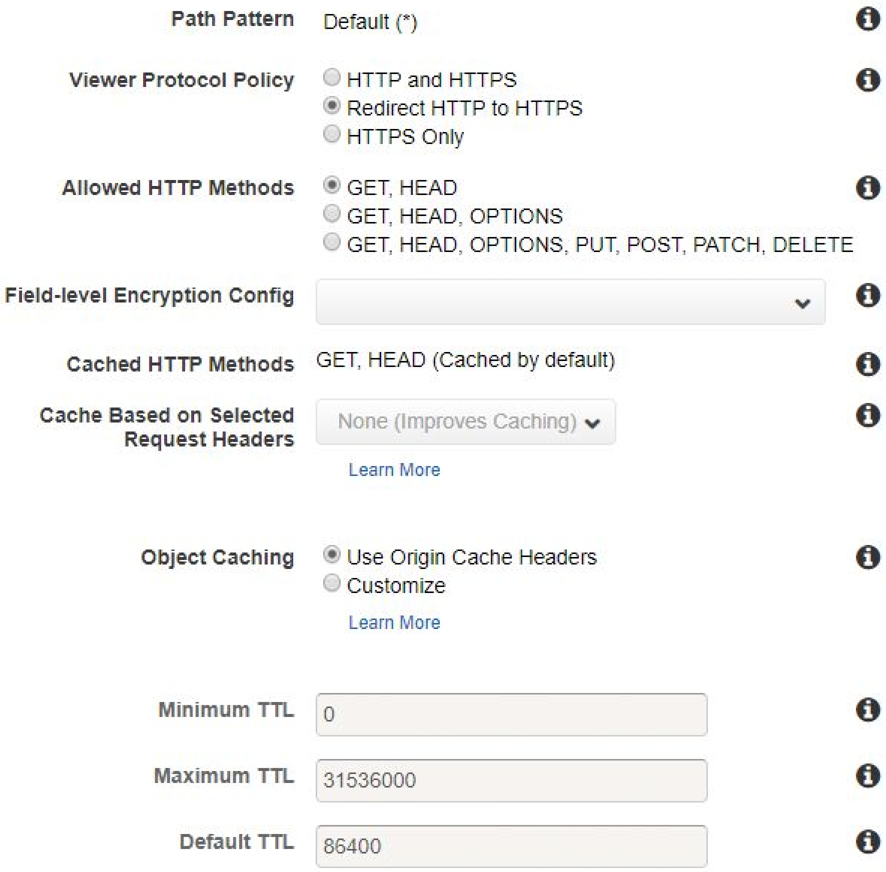
- Sign in to the AWS Management Console and in the **Find Services** search box type cloud and choose **CloudFront**.
- You should **Global** for the region at the top right.
- Click **Create Distribution**.
- Under **Web** click **Get Started**.
- For **Origin Domain Name** once you place the cursor in there you should see your available S3 buckets.
- Pick the website bucket you created.
- If it's not listed type it in: e.g `2019-03-01-er-website.s3.amazonaws.com` *Using your bucket name*
- Leave **Origin Path** blank.
- The **Origin ID** should have been pre-populated when you chose your bucket.
- Click **Yes** to **Restrict Bucket Access**.
- Under **Origin Access Identity** select **Create a New Identity**.
- It will pre-populate the **Comment** and append the bucket name.
- For **Grant Read Permissions on Bucket** check **Yes, Update Bucket Policy**. This will update the bucket policy for us.
- Leave the **Origin Custom Headers** blank.



Origin Settings

Origin Domain Name	2019-03-01-er-website.s3.amazonaws.c
Origin Path	
Origin ID	S3-2019-03-01-er-website
Restrict Bucket Access	● Yes ○ No
Origin Access Identity	● Create a New Identity ○ Use an Existing Identity
Comment	access-identity-2019-03-01-er-website.s
Grant Read Permissions on Bucket	● Yes, Update Bucket Policy ○ No, I Will Update Permissions
Origin Custom Headers	Header Name: (blank) Value: (blank)

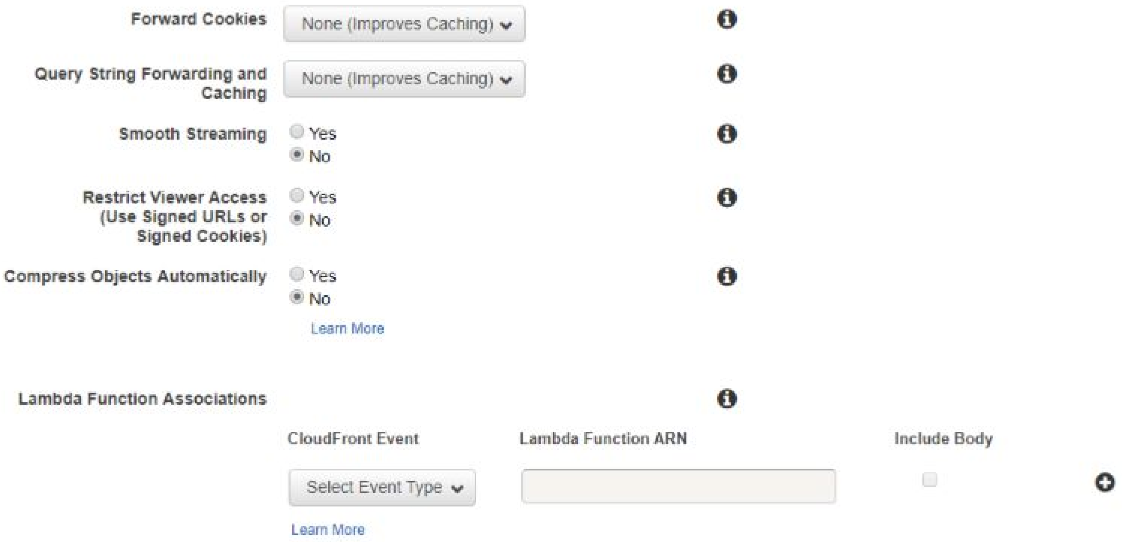
- For the **Default Cache Behavior Settings** section:
- Under **Viewer Protocol Policy** select **Redirect HTTP to HTTPS**.
- For **Allowed HTTP Methods** choose **GET, HEAD**.
- Leave **Field-level Encryption Config** blank.
- Leave **GET, HEAD (Cached by default)** for **Cached HTTP Methods**.
- For **Cache Based on Selected Request Headers** leave it as the default **None (Improves Caching)**.
- For **Object Caching** also leave it at the default **Use Origin Cache Headers**.



Path Pattern	Default (*)
Viewer Protocol Policy	○ HTTP and HTTPS ● Redirect HTTP to HTTPS ○ HTTPS Only
Allowed HTTP Methods	● GET, HEAD ○ GET, HEAD, OPTIONS ○ GET, HEAD, OPTIONS, PUT, POST, PATCH, DELETE
Field-level Encryption Config	
Cached HTTP Methods	GET, HEAD (Cached by default)
Cache Based on Selected Request Headers	None (Improves Caching) Learn More
Object Caching	● Use Origin Cache Headers ○ Customize Learn More
Minimum TTL	0
Maximum TTL	31536000
Default TTL	86400

- Under **Forward Cookies** leave it as **None (Improves Caching)**.
- Also for **Query String Forwarding and Caching** leave as **None (Improves Caching)**.
- For **Smoothing Streaming** select **No**.
- For **Restrict Viewer Access (Use Signed URLs or Signed Cookies)** select **No**.
- Also leave **Compress Objects Automatically** as **No**.

- We can also leave **Lambda Function Associations** as the default.



Forward Cookies: None (Improves Caching)

Query String Forwarding and Caching: None (Improves Caching)

Smooth Streaming: ○ Yes ● No

Restrict Viewer Access (Use Signed URLs or Signed Cookies): ○ Yes ● No

Compress Objects Automatically: ○ Yes ● No

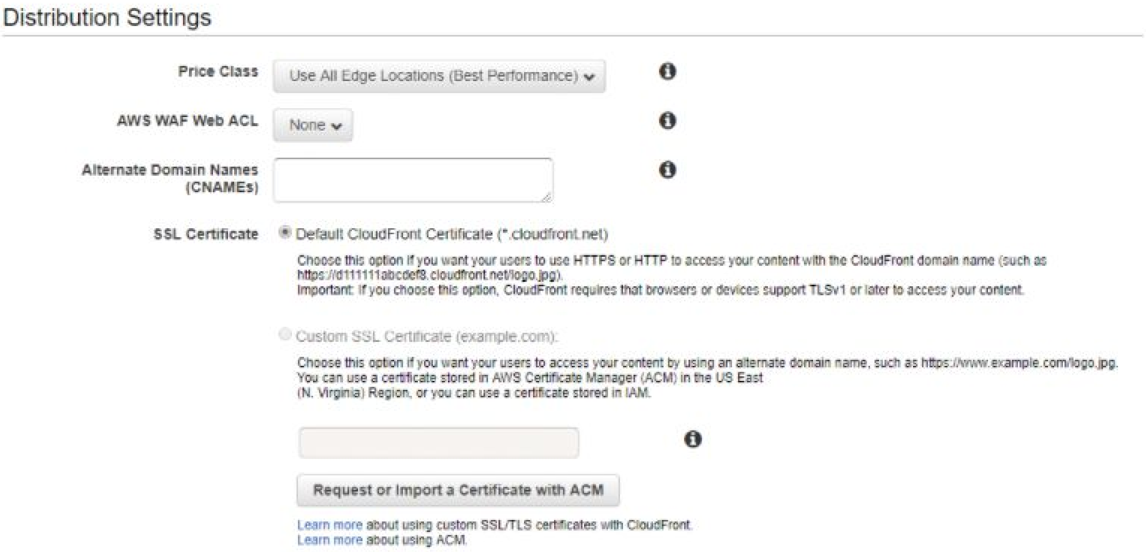
Learn More

Lambda Function Associations

CloudFront Event	Lambda Function ARN	Include Body
Select Event Type		□

Learn More

- Scroll down to **Distribution Settings**.
- For **Price Class** leave the default **Use All Edge Locations (Best Performance)**.
- We will not be using WAF so for **AWS WAF Web ACL** leave it as **None**.
- Also leave **Alternate Domain Names (CNAMEs)** blank.
- We will also use the **Default CloudFront Certificate** for **SSL Certificate**.



Distribution Settings

Price Class: Use All Edge Locations (Best Performance)

AWS WAF Web ACL: None

Alternate Domain Names (CNAMEs):

SSL Certificate: ● Default CloudFront Certificate (*.cloudfront.net)

Choose this option if you want your users to use HTTPS or HTTP to access your content with the CloudFront domain name (such as https://d111111abcdef8.cloudfront.net/logo.jpg).
Important: If you choose this option, CloudFront requires that browsers or devices support TLSv1 or later to access your content.

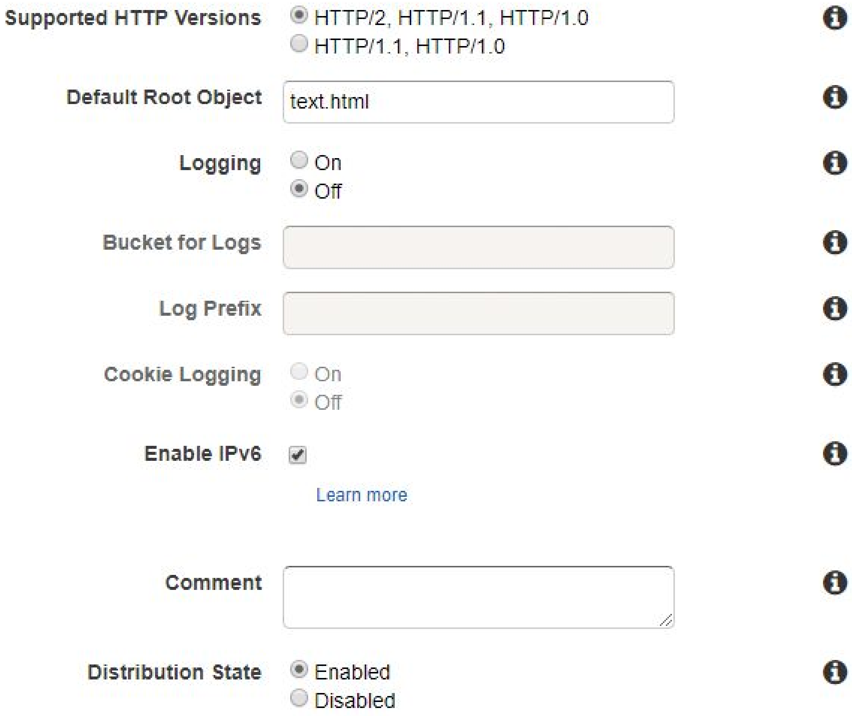
○ Custom SSL Certificate (example.com):

Choose this option if you want your users to access your content by using an alternate domain name, such as https://www.example.com/logo.jpg.
You can use a certificate stored in AWS Certificate Manager (ACM) in the US East
(N. Virginia) Region, or you can use a certificate stored in IAM.

Request or Import a Certificate with ACM

Learn more about using custom SSL/TLS certificates with CloudFront.
Learn more about using ACM.

- For **Supported HTTP Versions** leave as **HTTP/2, HTTP/1.1, HTTP/1.0**.
- Under **Default Root Object** type in `text.html`.
- We can leave **Logging** set to **Off**.
- Leave **Enable IPv6** checked.
- Finally set **Distribution State** to **Enabled**.



Supported HTTP Versions ● HTTP/2, HTTP/1.1, HTTP/1.0 ○ HTTP/1.1, HTTP/1.0

Default Root Object: text.html

Logging ○ On ● Off

Bucket for Logs:

Log Prefix:

Cookie Logging ○ On ● Off

Enable IPv6 ☑

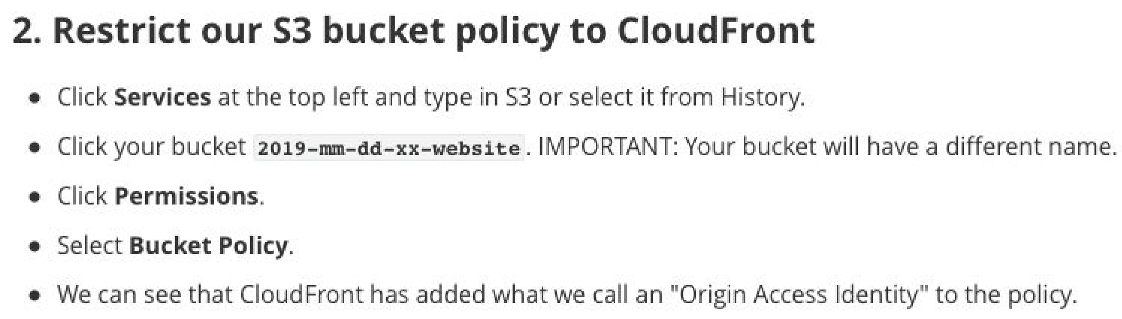
Learn more

Comment:

Distribution State ● Enabled ○ Disabled

- Click **Create Distribution**.
- Click on **Distributions** at the top left to see your CloudFront distribution being built.
- This can take 15-20 minutes to complete.

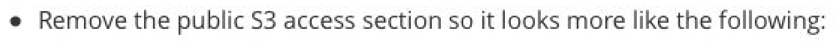
⏰ While we wait, we ill head over to S3 and lock down access to only allow calls from CloudFront.



2. Restrict our S3 bucket policy to CloudFront

- Click **Services** at the top left and type in S3 or select it from History.
- Click your bucket `2019-mm-dd-xx-website`. IMPORTANT: Your bucket will have a different name.
- Click **Permissions**.
- Select **Bucket Policy**.
- We can see that CloudFront has added what we call an "Origin Access Identity" to the policy.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "AddPerm",
         "Effect": "Allow",
         "Principal": "*",
         "Action": "s3:GetObject",
         "Resource": "arn:aws:s3:::2019-03-01-er-website/*"
      },
      {
         "Sid": "2",
         "Effect": "Allow",
         "Principal": {
            "AWS": "arn:aws:iam::cloudfront:user/CloudFront Origin Access Identity E1KO2GAPIWFF7X"
         },
         "Action": "s3:GetObject",
         "Resource": "arn:aws:s3:::2019-03-01-er-website/*"
      }
   ]
}
```



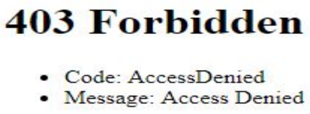
- Remove the public S3 access section so it looks more like the following:

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "2",
         "Effect": "Allow",
         "Principal": {
            "AWS": "arn:aws:iam::cloudfront:user/CloudFront Origin Access Identity
E1KO2GAPIWFF7X"
         },
         "Action": "s3:GetObject",
         "Resource": "arn:aws:s3:::2019-03-01-er-website/*"
      }
   ]
}
```

- This will only allow our specific CloudFront distribution access to our S3 bucket which is what we want.
- Click **Save** and grab a cup of coffee while we wait for the CloudFront Distribution to finish baking.

3. Steps for testing that we successfully locked down S3 from public view

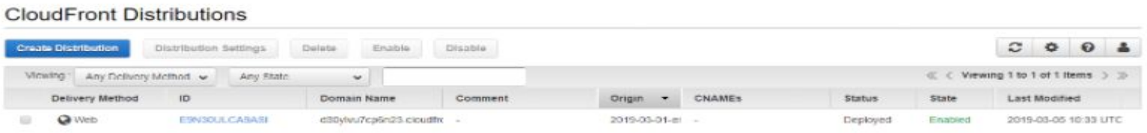
- Browse to **your** S3 endpoint: *Example:* *http://2019-03-01-er-website.s3-website-us-east-1.amazonaws.com/*
- You will see a **403 Forbidden** as we effectively removed public access via the bucket policy.



403 Forbidden

- Code: AccessDenied
- Message: Access Denied

- Click on the CloudFront distribution ID. (The blue hyperlink)



CloudFront Distributions

	Delivery Method	ID	Domain Name	Comment	Origin	CNAMEs	Status	State	Last Modified
□	Web	E9N3OULCA8ASI	d30ylvu7cp6n23.cloudfr	-	2019-03-01-er	-	Deployed	Enabled	2019-03-05 10:33 UTC

- Copy the URL under **Domain Name**.
- Browse to that URL and you should now see the **text.html** page.

⚠️ Remeber the distribution may take up to 15 minutes to complete.

Next we will wire up our static website to a backend API.

Awesome, we are moving though our exercise goal list nicely.