Delivering Content with CloudFront

Getting Ready for CloudFront

In this section, we will discover CloudFront with a demonstration.

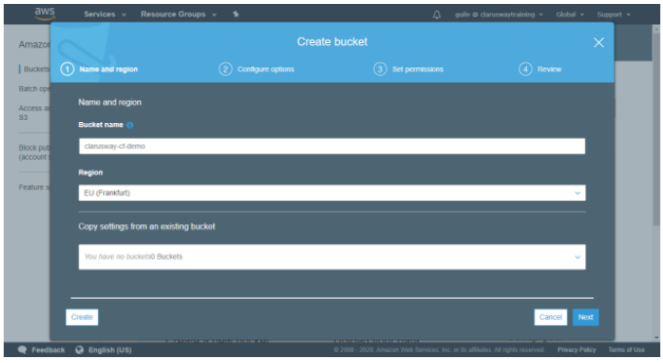
- First, we will create a bucket in S3, upload a sample image for our simple static web page,
- Then, we will configure CloudFront, setting our S3 bucket as origin,
- Finally, we will test our distribution to see whether CloudFront can deliver the content or not.

For the sake of this tutorial, we will use the CloudFront icon image (CF-icon-test.png) below. You can easily right click and save it for your sample web page or you can choose another image that you like.

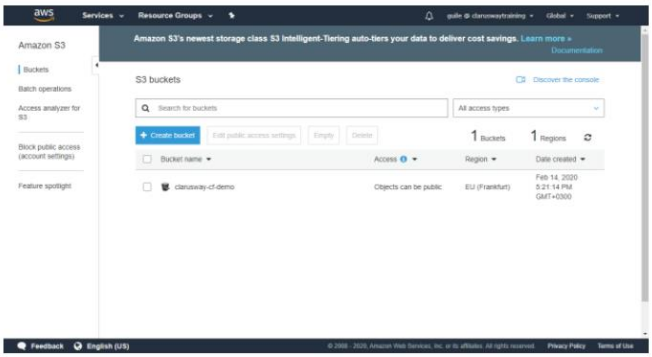
CF-icon-test.png

Setting S3 Bucket

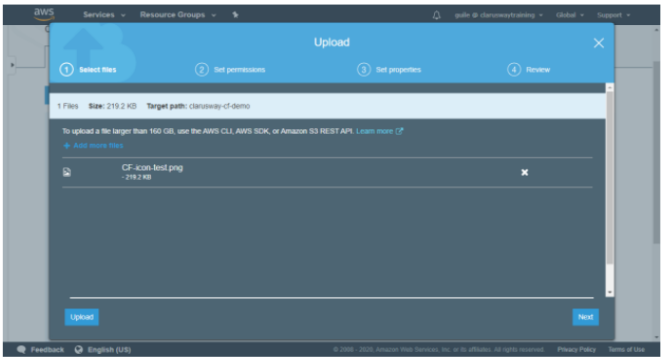
CloudFront is popular with S3 for static content like images, videos, HTML files etc. Thus, we will use an S3 bucket as origin for CloudFront in this demo. Let's login to the AWS Management Console, open S3 dashboard and click **Create bucket**.



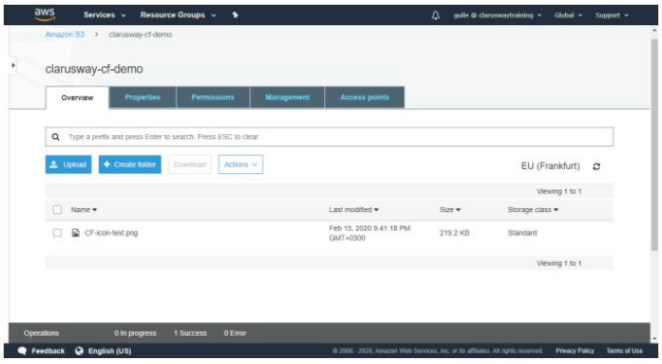
- Type a name for your bucket, choose a region and click Next. We will name our bucket "clarusway-cf-demo" for this tutorial.
- On **Configure options** tab leave all the options "as is", and click Next.
- Uncheck **Block all public access** to make it public on the **Set permissions** tab and acknowledge the warning about granting public access. Click **Next**.
- Check your settings on the **Review** tab and click **Create bucket**.



- Now that the bucket has been created, click the name of the bucket first, and then click **Upload**. Select **Add files** or drag-and-drop the image from your computer.



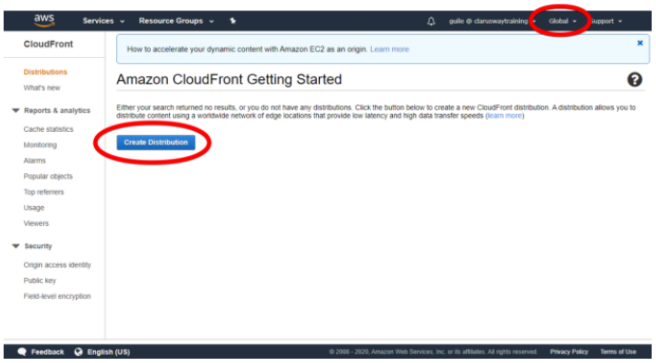
- Select **Grant public read access to this object(s)** on **Set permissions** tab to make it public.
- Choose your **Storage class**, click **Next**.
- Review your settings and click **Upload**.



As we are done on S3, let's move on to CloudFront.

Create Distribution

Type "CloudFront" to the service search box or go and directly click it under **Networking & Content Delivery** section. A distribution is needed to deliver the content via the network of edge locations. Hence, to get started with CloudFront, click **Create Distribution**.



Tip:

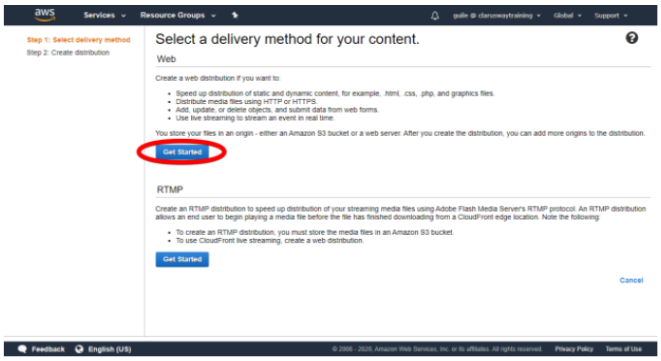
- Notice CloudFront is a global service and does not require a region selection.

Select Delivery Method

CloudFront has two delivery methods to distribute your content, a Web Distribution and an RTMP (Real-Time Messaging Protocol) Distribution.

- **Web**: Choose Web Distribution if you want CloudFront to,
  - Deliver your static and dynamic content like image files, .html and .css files etc.,
  - Distribute media files via HTTP or HTTPS,
  - Enable adding, updating, deleting objects and submitting data to web forms,
  - Stream live events in real time.
- **RTMP**: Choose RTMP if you want CloudFront to deliver your streaming media files via Adobe Flash Media Server's RTMP protocol. But, don't forget you must store the media files in an S3 bucket to use this method.

Due to the fact that the sample image, which will be delivered, is a static content, click **Get Started** under the Web option.



Web Distribution has 3 main settings that will be covered on the upcoming pages:

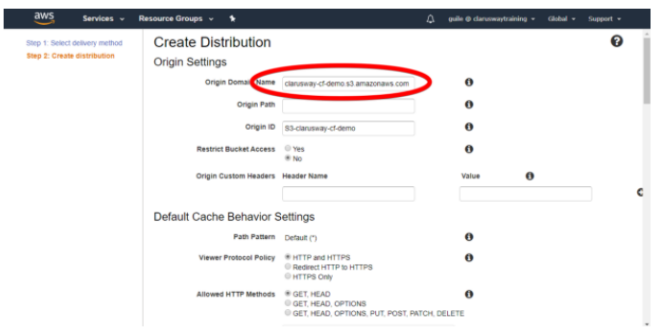
- Origin Settings
- Default Cache Behavior Settings
- Distribution Settings

Origin Settings

Origin Settings let users to define and customize their origins to be distributed.

- **Origin Domain Name**: Domain name for the origin is defined here. An S3 bucket, AWS MediaPackage channel endpoint, AWS MediaStore container endpoint, or a web server (like an ELB, EC2 or another server outside AWS) can be selected. Eligible resources will be automatically listed in the dropdown menu.
- **Origin Path**: Any other path for a directory under the origin is specified here. This setting is optional.
- **Origin ID**: A unique name must be described for the origin within the distribution. AWS will assign a default name when you select the origin domain name.
- **Restrict Bucket Access**: Enables or disables users to access S3 bucket only using CloudFront URL. Default value is "No".
- **Origin Custom Headers**: Custom headers can be specified here in key-value pairs. This setting is optional.

For this demo, let's select S3 bucket as origin for our distribution and leave the other settings "as is".

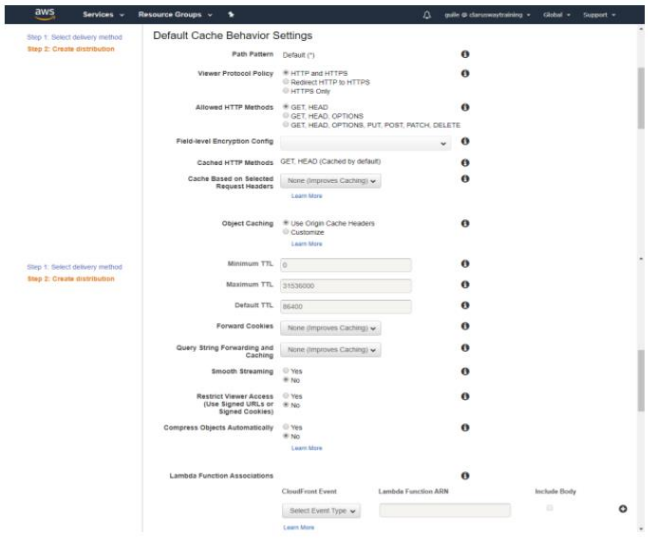


Default Cache Behavior Settings

Default Cache Behavior Settings let users to configure a variety of CloudFront functionality for a given URL path pattern for objects on the origin website.

- **Path Pattern**: Specifies which requests you want from this cache behavior to be applied. Comes with **Default (*)** (forward all requests). You have to add more cache behaviors after creating the distribution, to change the behavior or the routing for other requests.
- **Viewer Protocol Policy**: Lets you choose the protocol policy which is requested from viewers to use while accessing the content.
  - **HTTP and HTTPS**: Both protocols can be used.
  - **Redirect HTTP to HTTPS**: Both protocols can be used, but all HTTP requests will be redirected to HTTPS.
  - **HTTPS Only**: HTTP protocols can not be used, but only HTTPS.
- **Allowed HTTP Methods**: Lets you choose the HTTP methods which will be processed and forwarded to the origin.
  - GET, HEAD
  - GET, HEAD, OPTIONS
  - GET, HEAD, OPTIONS, PUT, POST, PATCH, DELETE
- **Field-level Encryption Config**: Lets you choose a field-level encryption configuration on specific data fields.
- **Cached HTTP Methods**: The response to GET and HEAD requests are always cached. The response to OPTIONS requests can be cached if selected here.
- **Cache Based on Selected Request Headers**: Enables you to select if CloudFront will cache objects based on the values of specified headers or not. Specific headers can be selected from a "Whitelist". "None" or "All" can be selected, as well.
- **Object Caching**: Lets you select the time that the objects will stay in the CloudFront cache. If the original server is using a Cache-control header to determine the TTL (Time-To-Live) of the objects which are set to the desired values, select **Use Origin Cache Headers**, else select **Customize** and set the desired TTL values.
- **Forward Cookies**: Enables you to select if CloudFront will forward cookies to the origin server or not. Desired cookies can be defined under a custom "Whitelist". "None" or "All" can be selected, as well.
- **Query String Forwarding and Caching**: Lets you select which query string parameters are desired to be forwarded to the origin (all or none) and which are desired to be based caching on (a whitelist of parameters or all).
- **Smooth Streaming**: Enables streaming live events in Microsoft Smooth Streaming format for servers not using Microsoft IIS (Internet Information Services).
- **Restrict Viewer Access**: Enables restricting viewers using Signed URLs or Signed Cookies to access the content.
- **Compress Objects Automatically**: Enables automatic content compressing for web requests including `Accept-Encoding:gzip` in the request header.
- **Lambda Function Associations**: Enables Lambda function ARNs to be associated with the specific event types in the dropbox.

For this demo, don't make any changes and keep the "default" settings.

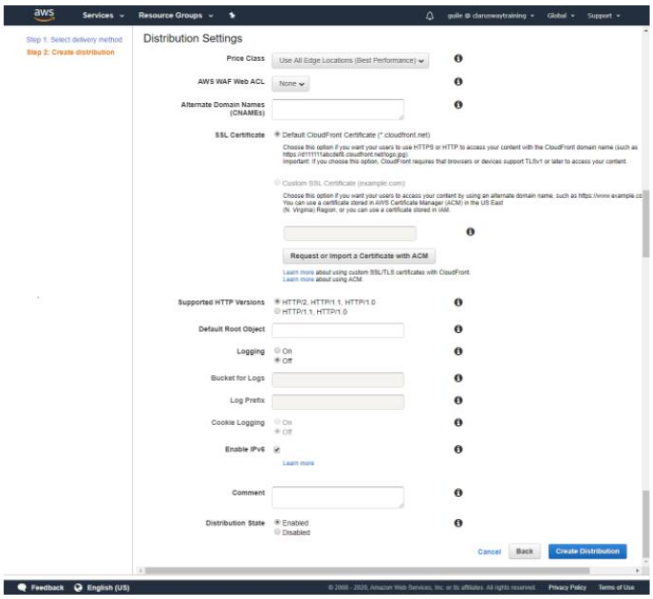


Distribution Settings

Distribution Settings let users to customize their distributions.

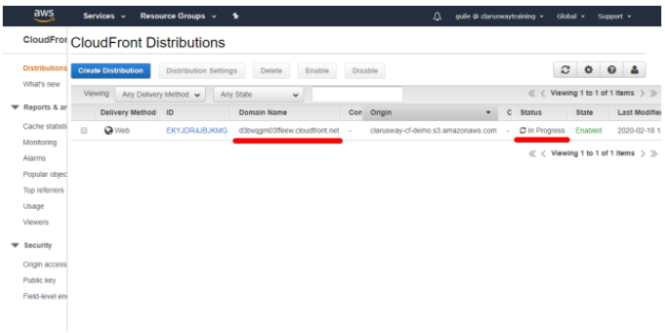
- **Price Class**: Enables selecting a group of edge locations out of three. Price is associated with the edge location group you need for your distribution. By default, CloudFront enables all edge locations for best performance.
- **AWS WAF Web ACL**: Enables using AWS WAF (Web Application Firewall) Web ACL (Access Control List) to block or allow predefined requests.
- **Alternate Domain Names (CNAMEs)**: Lets you specify your own domain name in the URLs for your objects in addition to your CloudFront domain name.
- **SSL Certificate**: Enables you to select the Default CloudFront Certificate or a custom SSL certificate.
- **Supported HTTP Versions**: Enables you to select the versions of the HTTP protocols which CloudFront will accept.
- **Default Root Object**: Enables you to specify the object that you want CloudFront to request from your origin when a user requests just the root URL of your distribution. Optional.
- **Logging**: Lets CloudFront log information about the viewer requests for the files in the distribution. After enabling, **Bucket for Logs**, **Log Prefix** and **Cookie Logging** can be set. Charging will occur for accesing logs.
- **Enable IPv6**: Enables CloudFront to accept IPv6 users as well as IPv4.
- **Comment**: Enables entering an optional comment about the distribution.
- **Distribution State**: Enables CloudFront to accept viewers requests or not.

Distribution Settings are the last settings to create a distribution. Keep the default settings for the demo and click **Create Distribution**.



Testing Distribution

We have just created a CloudFront distribution. Because of the wide network of CloudFront, it will take some time (approx. 15 mins.) for the status to change from *In Progress* to *Deployed* .

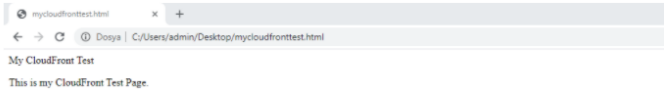


While the distribution is being deployed, let's prepare our test tool.

- Open a text editor on your computer.
- Copy the following HTML code and paste it to your editor.

```
<html>
<head>My CloudFront Test</head>
<body>
<p>This is my CloudFront Test Page.</p>
<p><img src="http://domain name/object name" alt="my test image"></p>
</body>
</html>
```

- Replace *domain name* with the domain name which CloudFront assigned to your distribution. In our case, it is *d3bvqgm03ffeew.cloudfront.net*.
- Replace *object name* with the name of your image file in the Amazon S3 bucket. In our case, it is *CF-icon-test.png*.
- Save the text file as an .html file, such as *mycloudfronttest.html*.
- If your distribution is deployed, open your web page in a browser.



If you can see the image on your web page, you have successfully set up a web distribution and delivered a static content with Amazon CloudFront, hosted in the cloud through Amazon S3. Go back and check your settings again if you're having problem displaying your content.