DuraCloud Media Streaming

HLS Streaming

HTTP Live Streaming (HLS) is an HTTP-based streaming protocol which supports delivery of adaptive bitrate video and audio streams. DuraCloud makes use of Amazon CloudFront to deliver its HLS streaming capability, so streaming content must be stored in the Amazon S3 storage provider to take advantage of this feature.

Media files must be transcoded into a format which is supported by HLS before streaming may occur. Typically transcoding results in a set of .ts segment files which contain the media in time-boxed fragments and .m3u8 index files which are used to capture the ordering and combinations of .ts files for playback.

Further background and information about HLS can be found on: wikipedia, encoding.com, and apple.com (as well as many other sites).

Examples

A simple example of displaying an HLS video file that is streamed via DuraCloud can be found on this page. If you review the page source, you will see that an HTML 5 video tag is used along with javascript from hls.js. The HTML 5 audio tag can be used similarly for audio content. The DuraCloud space where the video content is stored is set to allow open streaming (details on this below). The video in the example was transcoded using AWS Elastic Transcoder to create a single audio segment at 64k and a single video segment at 600k from a source MP4 video file.

Configuration of secure streaming is more involved due to the redirecting that is required (see the interaction flow below). For this reason, a demo application was created to provide an example of how your web application could utilize secure streams: https://github.com/duracloud/streaming-demo.

Open and Secure HLS Streaming

DuraCloud supports two different types of HLS streaming, open and secure. Open streaming allows anyone with access to the URL for a content item (in a space where open streaming is enabled) to stream that content. The exception to this is if the allowedOrigins parameter is utilized when enabling HLS streaming for a space, which limits browser-based (JavaScript) streaming requests to only certain hosts.

Secure streaming requires that permission be granted before a user is able to stream a file. Your web application will need to determine if the user has the necessary rights, and if so, allow a set of cookies to be set on that user's browser. Those cookies are used to verify that the user has permission to stream the files in a space. The request to generate those cookies can set further limits, such as the length of time the user will have access and the IP address range where the streaming is allowed to take place. The purpose of secure streaming is to restrict the use of the stream. This is ideal for scenarios where streamed content is not free to use or must only be provided to a limited audience.

Using HLS Streaming

Follow these steps to stream media files with DuraCloud using HLS

1. Create a space in DuraCloud which you will use to host streamed files
2. Transfer media files to the space. Be sure that the files have been transcoded to support HLS (.ts and .m3u8 files are expected)
3. Enable streaming
   a. To enable open streaming: Select the space in the DuraCloud interface and click the "ON" button next to "HLS Streaming:" in the Streaming section of the Space Detail pane. Alternatively, open streaming can be enabled using the same enable-streaming call as secure streaming, but with the "secure" parameter set to 'false".
   b. To enable secure streaming: Perform a POST HTTP call to the URL https://{institution}.duracloud.org/durastore/task/enable-streaming. The body of the POST request should include this JSON document: {"spaceId" : "","secure" : "true"}. Fill in the ID of the space to stream. Note that this call is using the DuraCloud REST API.
   c. For either open or secure streaming, if you would like to limit the hosts able to stream content, include the "allowedOrigins" parameter with a list of allowed origin values. For example, to allow calls from example.com and my.test.com, the value of allowedOrigins would be: ["https://example.com","https://my.test.com"]. Note that it is necessary to include the protocol as part of the origin, and that "http://" is considered different than "https://". The default allowedOrigins value is ["https://*"], which allows calls from all sites using an https protocol.
4. Wait up to 15 minutes. If this is the first time the space has been streamed, it can take up to 15 minutes for the files to be available on the Amazon edge servers.
5. Stream a file
   a. When using open streaming:
      i. Select an HLS index (playlist) file in the space. A video player will appear in the Content Detail pane. Verify that you are able to play the streamed file.
      ii. Look in the space properties for the HLS streaming host address. This is the path you will use for streaming files. Alternatively, you can perform a get-url-hls task call through the DuraCloud REST API to retrieve the streaming URL for each content item to be streamed. These URLs are predictable and do not expire.
   b. When using secure streaming:

        i. Spaces using secure streaming do not provide playback via the DuraCloud UI. You will need to perform a "get-signed-cookies-url" call, then redirect the user to the returned URL. More details about the get-signed-cookies-url call can be found in the Amazon S3 Storage Provider tasks section of the DuraCloud REST API.
6. Set up your website or application to provide access to the streamed files.

Secure HLS Streaming Interaction Flow

In order for media to be streamed from a space with secure streaming enabled, a set of cookies must first be set on the user's browser. These cookies provide the policy and signature used by Amazon CloudFront to verify that the user has permission to retrieve the files. One set of cookies applies to all content in a space. Each streamed space has its own streaming host and associated cookies.

Due to browser security requirements, cookies which apply to a domain can only be set in a response from that domain. In order to set cookies which apply to the CloudFront streaming domain, a call needs to be made to CloudFront by the user which results in the cookies being provided in the response. To make this happen, your web application (which is allowing the user to discover the streamed content) must interact with DuraCloud and the user following these steps:

1. Provide a way for the user to log in to your web application or otherwise verify that they should have access to streamed content (perhaps via IP)
2. On the next user request, your web application will make a call to the *get-signed-cookies-url* task in the DuraCloud REST API
    a. Set the *redirectUrl* parameter to the web address you want the user to be redirected to after cookies have been set (such as a list of files to stream).
    b. A *signedCookiesUrl* value will be returned from this call
3. Respond to the user's request with a 302 Redirect HTTP response, redirecting them to the *signedCookiesUrl* path
    a. This will result in the user being redirected to a CloudFront endpoint, which in turn makes a call to DuraCloud.
    b. The response to this call will include the necessary cookies and an HTML page with a refresh directive that will forward the user to the *re directUrl* you provided in the previous step.
4. Use the *get-url-hls* DuraCloud REST API call to retrieve the URL for each file to be streamed, then use an HLS viewer to stream the video. The user is allowed to see the video because of the cookies set on their browser in the previous steps.

The following diagram shows the flow of control described in the previous steps. The diagram uses a video list page as an example of a page the user could request from your web application, but this could be any page. The Web Application component is written and managed by the DuraCloud subscriber.

While this diagram may seem complex, the vast majority of the work is done for you by DuraCloud and CloudFront. As noted above, you need to make one call to retrieve the signed cookies URL, then respond to the user redirecting them to that URL. The rest of the flow for setting cookies is handled outside of your application. The user is then returned to your application in order to continue to search for media content, and that content is displayed in the same way as with open streaming.

Gliffy Macro Error

An error occurred while rendering this diagram. Please contact your administrator.

- **Name:** HLS Signed Streaming Flow

If you add files to a space with streaming turned on, those files will automatically be made available for streaming as well.