Send a stream from Wowza Streaming Engine to a Wowza CDN HLS target using Wowza REST APIs

Original Published on 05/17/2017 | Updated on 01/31/2020 9:26 am PST

Wowza Streaming Engine™ media server software can broadcast a live stream to a target in the Wowza Streaming Cloud™ service. This workflow allows you to bypass transcoding in Wowza Streaming Cloud and just use the Wowza Streaming Cloud edge network to deliver the stream to viewers.

You can achieve this workflow programmatically: First, use the Wowza Streaming Cloud REST API to create the HLS target to deliver the final stream to viewers. Then, use the Wowza Streaming Engine REST API to configure the default live application to send a single- or multi-bitrate stream to the Wowza Streaming Cloud HLS target.

Note: This workflow requires Wowza Streaming Engine 4.5 or later and a Wowza Streaming Cloud or Wowza CDN subscription.

Create the HLS stream target in Wowza Streaming Cloud

Using the Wowza Streaming Cloud REST API, create a Wowza CDN on Akamai stream target that receives the stream from Wowza Streaming Engine and delivers it to viewers using geographically distributed endpoints over HLS.

Wowza Streaming Cloud stream target parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>String</td>
<td>A descriptive name for the stream target, for example, MyCloudHLStarget.</td>
</tr>
<tr>
<td>provider</td>
<td>String</td>
<td>Specify akamai_cupertino.</td>
</tr>
</tbody>
</table>
Example request and response

**Notes:** In Wowza Streaming Cloud REST API code examples:

- To authenticate API requests, use **HMAC authentication** for production environments. For testing or proof of concept purposes only, use **API key and access key authentication**.
- The Wowza Streaming Cloud REST API curl examples below use environment variables. See [Using cURL](#) for more information on how to set these up.

Create the Wowza CDN on Akamai stream target:

```
curl -X POST \
  -H "Content-Type: application/json" \ 
  -H "wsc-api-key: ${WSC_API_KEY}" \ 
  -H "wsc-access-key: ${WSC_ACCESS_KEY}" \ 
  -d '{
    "stream_target_akamai": { 
      "name": "MyCloudHLStarget",
      "provider": "akamai_cupertino"
    }
  }' "${WSC_HOST}/api/${WSC_VERSION}/stream_targets/akamai"
```

The command creates the target and returns a response that looks something like this:

```
{
  "stream_target_akamai": {
    "id": "1234hjkl",
    "name": "MyCloudHLStarget",
    "provider": "akamai_cupertino",
    "use_secure_ingest": false,
    "use_cors": false,
    "stream_name": "46d95e9d",
    "primary_url": "http://post.[domain]-i.akamaihd.net/253398/46d95e9d",
    "hls_playback_url": "https://[domain]-i.akamaihd.net/hls/live/253398/46d95e9d/playlist.m3u8",
    "connection_code": "1effa1",
    "connection_code.expires_at": "2017-05-16T21:44:49.000Z",
    ...
  }
```

Related requests
View the details of a stream target, including its HLS playback URL:

curl -X GET \
-H "wsc-api-key: ${WSC_API_KEY}" \
-H "wsc-access-key: ${WSC_ACCESS_KEY}" \
"${WSC_HOST}/api/${WSC_VERSION}/stream_targets/akamai/[stream_target_id]/"

Delete a stream target:

curl -X DELETE \
-H "wsc-api-key: ${WSC_API_KEY}" \
-H "wsc-access-key: ${WSC_ACCESS_KEY}" \
"${WSC_HOST}/api/${WSC_VERSION}/stream_targets/akamai/[stream_target_id]"

Regenerate a target’s connection code:

curl -X PUT \
-H "wsc-api-key: ${WSC_API_KEY}" \
-H "wsc-access-key: ${WSC_ACCESS_KEY}" \
"${WSC_HOST}/api/${WSC_VERSION}/stream_targets/[stream_target_id]/regenerate_connection_code"

Create the stream source for the live application in Wowza Streaming Engine

Now, use the Wowza Streaming Engine REST API to configure the live application that ships with Wowza Streaming Engine to ingest a source stream (publisher).

Wowza Streaming Engine stream source (publisher) parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>publisherName</td>
<td>String</td>
<td>A descriptive name for the publisher, for example, MyRTMPencoder.</td>
</tr>
</tbody>
</table>

Example request and response

Note: Wowza Streaming Engine REST API requests must include three headers: Accept:application/json, Content-Type:application/json, and charset=utf-8. For more information, see Query the Wowza Streaming Engine REST API.
Create an RTMP source (publisher) for the **live** application on a local instance of Wowza Streaming Engine:

```bash
curl -X POST \
-H "Accept:application/json" \
-H "charset=utf-8" \
-H "Content-Type:application/json" \
-d '{
    "publisherName": "myRTMPencoder"
}'
"http://localhost:8087/v2/servers/_defaultServer_/vhosts/_defaultVHost_/applications/live/publishers/myRTMPencoder"
```

The command creates the publisher and returns a response that looks something like this:

```
{
    "success": true,
    "message": "",
    "data": null
}
```

Related requests

Get a list of applications:

```bash
curl -X GET \
-H "Accept:application/json" \
-H "charset=utf-8" \
-H "Content-Type:application/json" \
"http://localhost:8087/v2/servers/[serverName]/vhosts/[vhostName]/applications"
```

View the details of an application:

```bash
curl -X GET \
-H "Accept:application/json" \
-H "charset=utf-8" \
-H "Content-Type:application/json" \
"http://localhost:8087/v2/servers/[serverName]/vhosts/[vhostName]/applications/[appName]"
```

View a list of source streams (publishers) connected to an application:
Create the stream target for the live application in Wowza Streaming Engine

Still in the Wowza Streaming Engine REST API, create a stream target for the live application. The target will deliver the stream from Wowza Streaming Engine to the Wowza Streaming Cloud HLS target over HLS.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>connectionCode</td>
<td>String</td>
<td>The six-character \texttt{connection_code} from the Wowza Streaming Cloud HLS target, for example, \texttt{1effa1}.</td>
</tr>
<tr>
<td>enabled</td>
<td>Boolean</td>
<td>Specify \texttt{true} so that the target is ready to run as soon as you create it.</td>
</tr>
<tr>
<td>entryName</td>
<td>String</td>
<td>A descriptive name for the map entry, for example, \texttt{cloudHLStarget}.</td>
</tr>
<tr>
<td>profile</td>
<td>String</td>
<td>Specify \texttt{wowza\text{-}streaming\text{-}cloud}.</td>
</tr>
<tr>
<td>sourceStreamName</td>
<td>String</td>
<td>The name of the incoming stream for the live application, for example, \texttt{myStream}.</td>
</tr>
<tr>
<td>sourceStreamName</td>
<td>String</td>
<td>Determines whether Wowza Streaming Engine sends a single bitrate stream or a group of stream renditions to Wowza</td>
</tr>
</tbody>
</table>

```
curl -X GET \
-H "Accept:application/json" \
-H "charset=utf-8" \
-H "Content-Type:application/json" \
"http://localhost:8087/v2/servers/[servername]/vhosts/[vhostname]/applications/[appname]/publishers"
```
<table>
<thead>
<tr>
<th><strong>wowzaCloud.adaptiveStreaming</strong></th>
<th><strong>Boolean</strong></th>
<th>Streaming Cloud. Specify <strong>false</strong> to send a single bitrate stream, or specify <strong>true</strong> to send an adaptive bitrate stream as a group of stream renditions.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>wowzaCloudDestinationType</strong></td>
<td><strong>String</strong></td>
<td>Specify <strong>streamTarget</strong>.</td>
</tr>
</tbody>
</table>

**Example request and response**

```
curl -X POST \
-H "Accept:application/json" \
-H "charset=utf-8" \
-H "Content-Type:application/json" \
-d '{
  "connectionCode": "1effa1",
  "enabled": true,
  "entryName": "cloudHLStarget",
  "profile": "wowza-streaming-cloud",
  "sourceStreamName": "myStream",
  "wowzaCloud.adaptiveStreaming": "false",
  "extraOptions": {
    "wowzaCloudDestinationType": "streamTarget",
    "destinationName": "wowzastreamingcloud"
  }
}'
"http://localhost:8087/v2/servers/_defaultServer_/vhosts/_defaultVHost_/applications/live/pushpublish/mapentries/cloudHLStarget"
```

The command creates the target (map entry) and returns a response that looks something like this:

```
{
  "success": true,
  "message": "Entry (cloudHLStarget) saved successfully",
  "data": null
}
```

**Related requests**

View a list of stream targets (push publishing map entries):
View the details of a target (push publishing map entry) for an application on a local instance of Wowza Streaming Engine:

```
curl -X GET \
-H "Accept:application/json" \
-H "charset=utf-8" \
-H "Content-Type:application/json" \
"http://localhost:8087/v2/servers/[serverName]/vhosts/[vhostName]/applications/[appName]/pushpublish/mapentries"
```

Delete a stream target (push publishing map entry):

```
curl -X DELETE \
-H "Accept:application/json" \
-H "charset=utf-8" \
-H "Content-Type:application/json" \
"http://localhost:8087/v2/servers/[serverName]/vhosts/[vhostName]/applications/[appName]/pushpublish/mapentries/[entryName]"
```

Enable a single stream target (push publishing map entry):

```
curl -X PUT \
-H "Accept:application/json" \
-H "charset=utf-8" \
-H "Content-Type:application/json" \
"http://localhost:8087/v2/servers/[server_name]/vhosts/[vhostName]/applications/[appname]/pushpublish/mapentries/[entryname]/actions/enable"
```

Start streaming

Make sure your source is configured and connected to Wowza Streaming Engine (see [Connect a live source to Wowza Streaming Engine](#)). Then, start the source encoder or camera.