Send a stream from Wowza Streaming Engine to Wowza Streaming Cloud for transcoding using Wowza APIs

With Wowza Streaming Engine™ media server software version 4.2 and later and the Wowza Streaming Cloud™ service, you can send single bitrate streams from Wowza Streaming Engine to Wowza Streaming Cloud for adaptive bitrate transcoding and delivery to viewers over the Wowza CDN edge network. This workflow allows you to use Wowza Streaming Cloud’s high-performance servers and high-bandwidth networks for the heavy lifting of transcoding before delivering the stream to viewers.

In this article, learn how to use the Wowza Streaming Cloud REST API and the Wowza Streaming Engine REST API to set up a live application in Wowza Streaming Engine to send a stream to Wowza Streaming Cloud for transcoding.

Notes:
- Wowza Streaming Engine 4.7.4 has a known error that may break the stream target functionality for sending streams to Wowza Streaming Cloud for transcoding. We recommend updating to Wowza Streaming Engine 4.7.5 or later. For more information, see Wowza Streaming Engine 4.7.4 may experience problems sending streams to Wowza Streaming Cloud for transcoding.
- For a UI-based version of this workflow, see Send a live stream from Wowza Streaming Engine to Wowza Streaming Cloud for transcoding.

Create a live stream in Wowza Streaming Cloud

1. Create a live stream using the Wowza Streaming Cloud REST API that will receive the stream from Wowza Streaming Engine, transcode it, and deliver it to viewers using the Wowza CDN.

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.

The following request creates a live stream that uses Wowza Streaming Engine as the source. Set the `delivery_type` to **single-bitrate** to send a single-bitrate stream to Wowza Streaming Cloud for transcoding. For information about available parameters, see the **Wowza Streaming Cloud REST API reference documentation**.

```
curl -X POST 
-H "Content-Type: application/json" 
-H "wsc-api-key: ${WSC_API_KEY}" 
-H "wsc-access-key: ${WSC_ACCESS_KEY}" 
-d '{
   "live_stream": {
      "aspect_ratio_height": 720,
      "aspect_ratio_width": 1280,
      "billing_mode": "pay_as_you_go",
      "broadcast_location": "us_west_california",
      "encoder": "wowza_streaming_engine",
      "delivery_type": "single-bitrate",
      "transcoder_type": "transcoded",
      "name": "Mylivestream"
   }
}' 
"${WSC_HOST}/api/${WSC_VERSION}/live_streams"
```

The response includes information about the live stream, including the connection code, which you’ll need to configure Wowza Streaming Engine to send the stream to Wowza Streaming Cloud. The response should look something like this:
Related requests

View the details of a live stream:

```json
{
    "live_stream": {
        "id": "abcrl8v5",
        "name": ...
    },
    "created_at": "2018-08-31T17:47:58.000Z",
    "updated_at": "2018-08-31T17:47:59.000Z"
}
```
Create a live application in Wowza Streaming Engine

Next, 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 source stream (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, <em>MyRTMPencoder</em>.</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:
Create a stream target in Wowza Streaming Engine

Still in the Wowza Streaming Engine REST API, create a stream target for the **live** application.

**Wowza Streaming Engine map entries parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>connectionCode</code></td>
<td>String</td>
<td>The six-character <em>connection_code</em> from the Wowza Streaming Cloud live stream, for example, <em>0oUm9G</em>.</td>
</tr>
<tr>
<td><code>enabled</code></td>
<td>Boolean</td>
<td>Determines whether the stream target is enabled when it's created. Specify <em>true</em> so that the target is ready to run as soon as you create it. If you don't include this parameter, you'll need a separate API request to enable the stream target.</td>
</tr>
<tr>
<td><code>entryName</code></td>
<td>String</td>
<td>A descriptive name for the map entry, for example, <em>cloudLiveStream</em>.</td>
</tr>
</tbody>
</table>

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

```
{
  "success": true,
  "message": 
```
**Parameter** | **Data Type** | **Description**
--- | --- | ---
profile | String | defines how the stream is sent to its destination and determines which additional configurable parameters are available. Specify `wowza-streaming-cloud`.

**sourceStreamName** | String | The name of the incoming stream for the live application, for example, `myStream`.

**wowzaCloudDestinationType** | String | Specify `transcoder`.

**wowzaCloud.adaptiveStreaming** | Boolean | Determines whether Wowza Streaming Engine sends a single bitrate stream or a group of stream renditions to Wowza Streaming Cloud. Specify `false` to send a single bitrate stream.

---

**Example request and response**

```
curl -X POST -H "Accept:application/json" -H "charset=utf-8" -H "Content-Type:application/json" -d '{
  "connectionCode": "0oUm9G ",
  "enabled": true,
  "entryName": "cloudLiveStream",
  "profile": "wowza-streaming-cloud",
  "sourceStreamName": "myStream",
  "wowzaCloud.adaptiveStreaming": "false",
  "extraOptions": {
    "wowzaCloudDestinationType": "transcoder",
    "destinationName": "wowzastreamingcloud"
  }
}' "http://localhost:8087/v2/servers/_defaultServer_/vhosts/_defaultVHost_/applications/live/pushpublish/mapentries/cloudLiveStream"
```

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

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

With the live application in Wowza Streaming Engine configured to ingest source stream and deliver it to Wowza Streaming Cloud, you’re ready to test your workflow.

1. Configure your source encoder or camera. See Connect a live source to Wowza Streaming Engine for more information.

2. Start the live stream using the Wowza Streaming Cloud REST API:

3. Fetch the state of the live stream to make sure it’s started:

4. Start the stream in the H.264 camera or encoder that’s sending the stream to the live application in Wowza Streaming Engine.

5. Confirm that the stream is playing.
   a. Fetch a video thumbnail of the stream using the Wowza Streaming Cloud REST API:

   b. View the thumbnail URL in a browser.

   c. Alternatively, in the Wowza Streaming Cloud web manager, confirm that the live stream is playing by looking at the Video Thumbnail in the Overview tab of the live stream detail page.

6. Use the Wowza Streaming Cloud REST API to stop the live stream. Alternatively, click Stop Stream at the top of the live stream detail page in the Wowza Streaming Cloud web manager.

Stop the live stream:
7. Stop the stream in the source camera or encoder.