Connect a UDP encoder to Wowza Streaming Cloud using the Wowza Streaming Cloud REST API

The Wowza Streaming Cloud™ service can connect to any H.264 encoder that supports the User Datagram Protocol (UDP). UDP is a connectionless protocol that makes it easier to transmit information quickly, but is also more prone to network issues, packet loss, and packets arriving out of order.

**Note:** Source authentication isn't available for UDP.

Create a live stream

The live stream workflow allows you to configure more settings in one API request, while the transcoder workflow allows more modular, custom configuration of settings using multiple API requests. Start with this section if you choose the live stream workflow.

1. Using the Wowza Streaming Cloud REST API, create a live stream.

### Live stream parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>aspect_ratio_height</td>
<td>integer</td>
<td>The height, in pixels, of the output rendition. This value should correspond to the aspect ratio (widescreen or standard) of the video source and be divisible by 8. Set the aspect ratio of the live stream to match the aspect ratio in your encoder settings.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Data Type</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>-----------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>aspect_ratio_width</td>
<td>integer</td>
<td>The width, in pixels, of the output rendition. This value should correspond to the aspect ratio (widescreen or standard) of the video source and be divisible by 8. Set the aspect ratio of the live stream to match the aspect ratio in your encoder settings.</td>
</tr>
<tr>
<td>billing_mode</td>
<td>string</td>
<td>The billing mode for the stream. Specify the default value, <strong>pay_as_you_go</strong>.</td>
</tr>
<tr>
<td>broadcast_location</td>
<td>string</td>
<td>The region that’s closest to where your stream originates. For a list of valid regions, see the API reference documentation.</td>
</tr>
<tr>
<td>encoder</td>
<td>string</td>
<td>The video source for the live stream. Specify <strong>other_udp</strong>.</td>
</tr>
<tr>
<td>name</td>
<td>string</td>
<td>The name of the live stream. Enter an alphanumeric string that is short (maximum 200 characters) and descriptive, for example, <strong>MyUDPStream</strong>.</td>
</tr>
<tr>
<td>transcoder_type</td>
<td>string</td>
<td>The type of transcoder. Specify the default value, <strong>transcoded</strong>. You can alternatively use <strong>passthrough</strong>, depending on your needs and the functionality available at your broadcast location.</td>
</tr>
<tr>
<td>delivery_method</td>
<td>string</td>
<td>The method you’re using to deliver the source stream to the transcoder. Specify the default value, <strong>push</strong>. Wowza Streaming Cloud doesn’t support <strong>pull</strong> or <strong>cdn</strong> connections for UDP.</td>
</tr>
</tbody>
</table>
For a full list of live stream parameters, see Live Streams in the Wowza Streaming Cloud API reference.

Example request and response

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

This request creates a live stream with an \textit{id} parameter, an associated player, and a hosted page. The details of the live stream's configuration are listed in the response. For UDP, you'll use the \textit{source_connection_information} to configure the stream.

```json
{
   "live_stream": {
      "id": "1234abcd",
      "name": "MyUDPStream",
      ...
      "encoder": "other_udp",
      ...
      "source_connection_information": {
         "primary_server": "udp://[subdomain].entrypoint.cloud.wowza.com",
         "host_port": "10000",
         "stream_name": "54g2813p"
         "disable_authentication": "true"
         "username": "null"
         "password": "null"
      },
      ...
   }
}
```
Related API requests

View the details of a live stream, including the player embed code and hosted page URL:

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

Update a live stream’s configuration:

curl -X PATCH \
-H "Content-Type: application/json" \
-H "wsc-api-key: ${WSC_API_KEY}" \
-H "wsc-access-key: ${WSC_ACCESS_KEY}" \
-d '{  "live_stream": {  "name": "MyDifferentLiveStreamName"  }}' "${WSC_HOST}/api/${WSC_VERSION}/live_streams/[live_stream_id]"

Delete a live stream:

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

2. Next, configure the UDP encoder to send the source stream to Wowza Streaming Cloud. To publish a test stream to Wowza Streaming Cloud, see Test the live stream connection.

Configure the source for the live stream

For a hardware encoder, make sure you have the latest firmware installed. See the encoder’s user guide for details about how to operate the device or software and how to specify settings such as resolution, bitrate, and frame rate.

1. Use the primary_server and host_port values returned in the REST API response to configure the UDP encoder.

2. Refer to documentation for your specific encoder to note where to input the
stream settings.

For example, **Address** is the *primary_server* value:

[wowza_subdomain].entrypoint.cloud.wowza.com

While **Destination Port** is the *host_port* value: 10000

**Test the live stream connection**

1. Use the *primary_server* and *host_port* from the REST API response to configure the stream settings.
2. Start the live stream using the Wowza Streaming Cloud REST API.

**Start the live stream:**

```bash
curl -X PUT \
-H "wsc-api-key: ${WSC_API_KEY}" \
-H "wsc-access-key: ${WSC_ACCESS_KEY}" \
"${WSC_HOST}/api/${WSC_VERSION}/live_streams/[live_stream_id]/start"
```

Alternatively, click **Start Live Stream** at the top of the live stream detail page in the Wowza Streaming Cloud user interface.

3. If you’re using the Wowza Streaming Cloud REST API to start the live stream, fetch the state of the live stream to make sure it’s started.

**Fetch the state of the live stream:**

```bash
curl -X GET \
-H "wsc-api-key: ${WSC_API_KEY}" \
-H "wsc-access-key: ${WSC_ACCESS_KEY}" \
"${WSC_HOST}/api/${WSC_VERSION}/live_streams/[live_stream_id]/state"
```

4. Confirm that the stream is playing.

   a. Fetch a video thumbnail of the stream using the Wowza Streaming Cloud REST API.

**Fetch the thumbnail URL of a live stream:**

```bash
curl -X GET \
-H "wsc-api-key: ${WSC_API_KEY}" \
-H "wsc-access-key: ${WSC_ACCESS_KEY}" \
"${WSC_HOST}/api/${WSC_VERSION}/live_streams/[live_stream_id]/thumbnail_url"
```
b. View the thumbnail URL in a browser.
c. Alternatively, in the Wowza Streaming Cloud user interface, confirm that the live stream is playing by looking at the **Video Thumbnail** in the **Overview** tab of the live stream detail page.

5. Use the Wowza Streaming Cloud REST API to stop the live stream.

**Stop the live stream:**

```
curl -X PUT \
-H "wsc-api-key: ${WSC_API_KEY}" \
-H "wsc-access-key: ${WSC_ACCESS_KEY}" \
"${WSC_HOST}/api/${WSC_VERSION}/live_streams/[live_stream_id]/stop"
```

Alternatively, click **Stop Live Stream** at the top of the live stream detail page in the Wowza Streaming Cloud user interface.

**Create a transcoder**

1. Using the Wowza Streaming Cloud REST API, create a transcoder.

**Transcoder parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>billing_mode</td>
<td>string</td>
<td>The billing mode for the stream. Use the default, <strong>pay_as_you_go</strong>.</td>
</tr>
<tr>
<td>broadcast_location</td>
<td>string</td>
<td>Specify the region that’s closest to where your stream originates. For a list of valid regions, see the <strong>API reference documentation</strong>.</td>
</tr>
<tr>
<td>delivery_method</td>
<td>string</td>
<td>The method you’re using to deliver the source stream to the transcoder. Specify <strong>push</strong>. Wowza Streaming Cloud doesn’t support <strong>pull</strong> or <strong>cdn</strong> connections for UDP.</td>
</tr>
<tr>
<td>name</td>
<td>string</td>
<td>The name of the transcoder. Enter an alphanumeric string that is short (maximum 200 characters) and descriptive, for example,</td>
</tr>
</tbody>
</table>
### Parameter Data Type Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>protocol</strong></td>
<td>string</td>
<td>The transport protocol you’re using to send the encoded stream to the transcoder. Use <strong>udp</strong>.</td>
</tr>
<tr>
<td><strong>transcoder_type</strong></td>
<td>string</td>
<td>Specify the default, <strong>transcoded</strong>. You can alternatively use <strong>passthrough</strong>, depending on your needs and the functionality available at your broadcast location.</td>
</tr>
</tbody>
</table>

### Example request and response

```bash
curl -X POST \  
-H "Content-Type: application/json" \  
-H "wsc-api-key: ${WSC_API_KEY}" \  
-H "wsc-access-key: ${WSC_ACCESS_KEY}" \  
-d '{"transcoder": {
   "billing_mode": "pay_as_you_go",
   "broadcast_location": "us_west_california",
   "delivery_method": "push",
   "name": "MyUDPTranscoder",
   "protocol": "udp",
   "transcoder_type": "transcoded"
 }}' "${WSC_HOST}/api/${WSC_VERSION}/transcoders"
```

This request creates a transcoder with an *id* parameter, but no outputs or stream targets. The details of the transcoder’s configuration are listed in the response, which should look something like this:
2. Complete the transcoder by adding output renditions and stream targets. For instructions, see one of the following articles, depending on whether you’re creating an adaptive bitrate or passthrough transcoder:

- Create an ABR stream and send it to a target with the Wowza Streaming Cloud REST API
- Pass a stream through the transcoder to a target with the Wowza Streaming Cloud REST API

3. Next, configure the source encoder. See Configure the source for next steps.

**Configure the source for the transcoder**

Next configure the UDP encoder to send the source stream to Wowza Streaming Cloud.

For a hardware encoder, make sure you have the latest firmware installed. See the encoder’s user guide for details about how to operate the device or software and how to specify settings such as resolution, bitrate, and frame rate.

1. Use the `domain_name` and `source_port` values returned when you created the transcoder to configure the UDP encoder.

2. Refer to documentation for your specific encoder to note where to input the stream settings.

   For example, **Address** is the `domain_name` value:
<table>
<thead>
<tr>
<th>Test the transcoder connection</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1.</strong> Start the transcoder using the Wowza Streaming Cloud REST API.</td>
</tr>
<tr>
<td><strong>Start the transcoder:</strong></td>
</tr>
<tr>
<td>curl -X PUT \</td>
</tr>
<tr>
<td>-H &quot;wsc-api-key: $(WSC_API_KEY)&quot; \</td>
</tr>
<tr>
<td>-H &quot;wsc-access-key: $(WSC_ACCESS_KEY)&quot; \</td>
</tr>
<tr>
<td>&quot;${WSC_HOST}/api/${WSC_VERSION}/transcoders/[transcoder_id]/start&quot;</td>
</tr>
<tr>
<td>Alternatively, click <strong>Start Transcoder</strong> at the top of the transcoder detail page in the Wowza Streaming Cloud user interface.</td>
</tr>
<tr>
<td><strong>2.</strong> If you’re using the Wowza Streaming Cloud REST API to start the transcoder, fetch the state of the transcoder to make sure it’s started.</td>
</tr>
<tr>
<td><strong>Fetch the state of the transcoder:</strong></td>
</tr>
<tr>
<td>curl -X GET \</td>
</tr>
<tr>
<td>-H &quot;wsc-api-key: $(WSC_API_KEY)&quot; \</td>
</tr>
<tr>
<td>-H &quot;wsc-access-key: $(WSC_ACCESS_KEY)&quot; \</td>
</tr>
<tr>
<td>&quot;${WSC_HOST}/api/${WSC_VERSION}/transcoders/[transcoder_id]/state&quot;</td>
</tr>
<tr>
<td><strong>3.</strong> Start the stream in the UDP encoder.</td>
</tr>
<tr>
<td><strong>4.</strong> Confirm that the stream is playing.</td>
</tr>
<tr>
<td><strong>a.</strong> Fetch a video thumbnail of the stream using the Wowza Streaming Cloud REST API.</td>
</tr>
<tr>
<td><strong>Fetch the thumbnail URL of a transcoder:</strong></td>
</tr>
<tr>
<td>curl -X GET \</td>
</tr>
<tr>
<td>-H &quot;wsc-api-key: $(WSC_API_KEY)&quot; \</td>
</tr>
<tr>
<td>-H &quot;wsc-access-key: $(WSC_ACCESS_KEY)&quot; \</td>
</tr>
<tr>
<td>&quot;${WSC_HOST}/api/${WSC_VERSION}/transcoders/[transcoder_id]/thumbnail_url&quot;</td>
</tr>
<tr>
<td><strong>b.</strong> View the thumbnail URL in a browser.</td>
</tr>
<tr>
<td><strong>c.</strong> Alternatively, in the Wowza Streaming Cloud user interface, confirm that</td>
</tr>
</tbody>
</table>

While **Destination Port** is the **source_port** value: 10000
the transcoder is playing by looking at the **Video Thumbnail** in the **Overview** tab of the transcoder detail page.

5. Use the Wowza Streaming Cloud REST API to stop the transcoder.

**Stop the transcoder:**

```bash
curl -X PUT  
-H "wsc-api-key: ${WSC_API_KEY}"  
-H "wsc-access-key: ${WSC_ACCESS_KEY}"  
"${WSC_HOST}/api/${WSC_VERSION}/transcoders/[transcoder_id]/stop"
```

Alternatively, click **Stop Transcoder** at the top of the transcoder detail page in the Wowza Streaming Cloud user interface.

6. Stop the stream in the source camera or UDP encoder.

**More resources**

- [Wowza Streaming Cloud REST API reference documentation](#)
- [Connect a UDP encoder to Wowza Streaming Cloud](#)