Linux Academy

Hands-On Training

Streaming Video with Elastic Transcoder, S3, and CloudFront

Contents

Introduction

With AWS, streaming video to your website or app can be done through the use of the **Elastic Transcoder**, **S3**, and **CloudFront**. In this lab, we are combining our knowledge of these services to upload a video, create a pipeline to transcode the video, and output it to S3. We are also configuring a CloudFront distribution to stream the video to a website.

We should be able to upload a video to an initial S3 bucket, which then, using an Elastic Transcoder job, transcodes the video. The Transcoder then dumps the output into a separate bucket, from which we can stream.

Getting Started

Log into the **AWS Console** using the credentials provided on the Live! Lab page. Additionally, download either the small or large **test video** available on the Live! Lab page. Those with limited bandwidth should download the smaller video.

Creating the Buckets

Navigate to the **S3 Dashboard**, and select **Create Bucket**. Your bucket name must be unique across all of the S3 namespace. We named the bucket in our example *linuxacademy-lab-input*, but yours needs to be different. Change the **Region** to *US Standard*. Press **Create**.

Repeat this process to create a second bucket that we called *linuxacademy-lab-output*. Again, your bucket name needs to be different due to S3 namespace requirements.

Upload the video provided to your *input* bucket. You may continue the rest of the lab while it downloads.

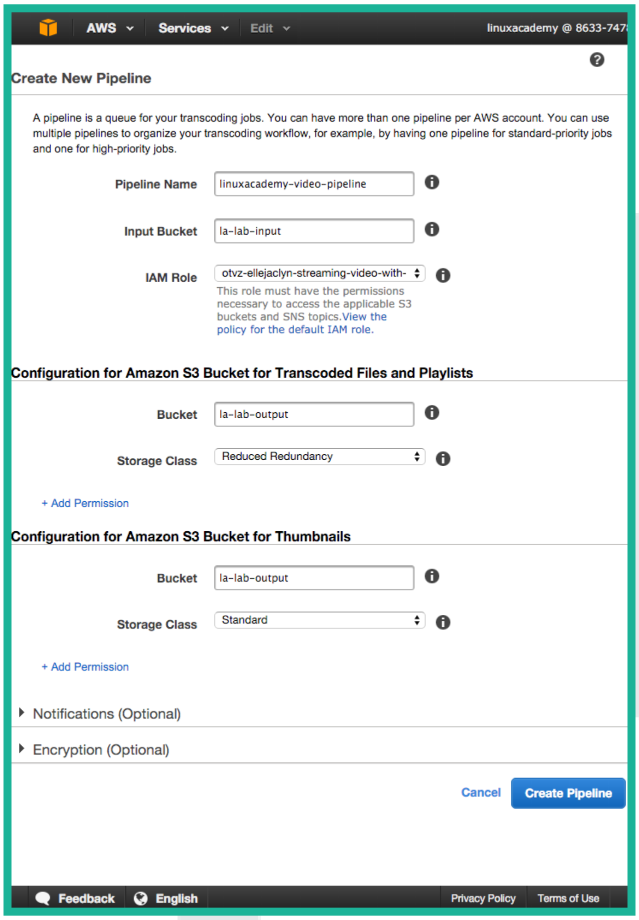
Creating a CloudFront Distribution

Open up your **CloudFront Dashboard**. Press **Create Distribution**. We now need to choose whether we are using a *web distribution* or an *RTMP distribution* for streaming. While you may feel, given the nature of this lab, that an RTMP is a proper way to go, we need the *Web* instance, which is not reliant on Flash protocols and allowed HTML5 playback. Select the web distribution.

The **Origin Domain Name** is the bucket from which we are streaming. As such, set this to the *output* bucket. The rest of the options can be left at default; more information can be found in the lab video in regards to each setting, but is outside the scope of this written guide.

CloudFront can take up to fifteen minutes to propagate. Continue to the next section of this lab as you wait.

Creating the Elastic Transcoder



Create New Pipeline

A pipeline is a queue for your transcoding jobs. You can have more than one pipeline per AWS account. You can use multiple pipelines to organize your transcoding workflow, for example, by having one pipeline for standard-priority jobs and one for high-priority jobs.

Pipeline Name: linuxacademy-video-pipeline

Input Bucket: la-lab-input

IAM Role: otvz-ellejaclyn-streaming-video-with-
This role must have the permissions necessary to access the applicable S3 buckets and SNS topics. View the policy for the default IAM role.

Configuration for Amazon S3 Bucket for Transcoded Files and Playlists

Bucket: la-lab-output

Storage Class: Reduced Redundancy

+ Add Permission

Configuration for Amazon S3 Bucket for Thumbnails

Bucket: la-lab-output

Storage Class: Standard

+ Add Permission

Notifications (Optional)

Encryption (Optional)

We want to be able to upload a file and have the Elastic Transcoder transcode and upload this file to the output bucket. While this process is something that can be done through automation in most practical environments, we are doing it manually to gain a better understanding of each step.

So how do we take our MP4 video and re-encode it for lower resolutions? We use AWS's Elastic Transcoder.

Open the **Elastic Transcoder Dashboard**, found under **Application Services**. Press **Create a new Pipeline**. Pipelines allow us to create a job and then queues that job on the server for processing.

We named our pipeline *linuxacademy-video-lab* and selected our *input* bucket for the **Input Bucket** value. Select the **IAM Role** that is already available. It should contain your username.

Under **Configuration for Amazon S3 Bucket for Transcoded Files and Playlists**, select the *output* bucket for **Bucket**. For **Storage Class** select *Reduced Redundancy*, since these files are easily reproducible from the originals in the input bucket. For thumbnail options, choose the *output* bucket, and use *Standard* storage. In this instance, standard storage is appropriate because we do not want the Elastic Transcoder to re-transcode a video based upon a missing thumbnail — this would be an unnecessary excess cost.

Notifications and **Encryption** can be left with default selections. Press **Create Pipeline**.

Creating a Job

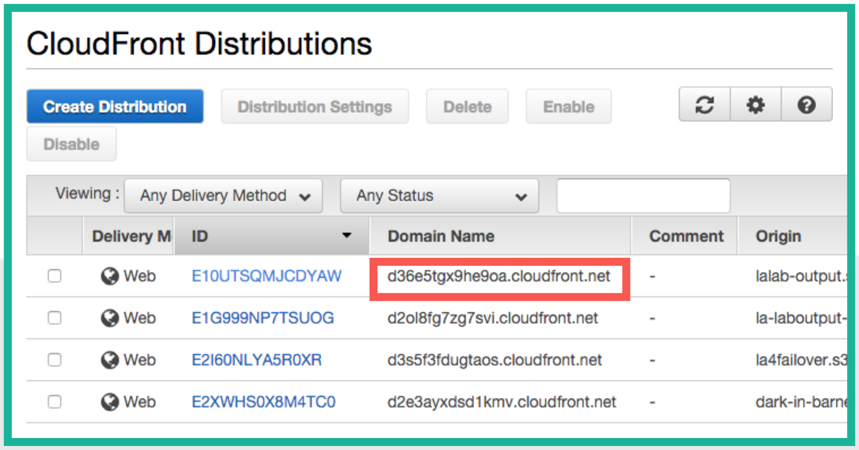
A job is a task that the Elastic Transcoder needs to perform. For this lab, that is taking the video we have uploaded and transcoding it. Press **Create New Job**.

Select the *linuxacademy-video-pipeline* from the **Pipeline** drop-down list. For **Input Key**, choose the uploaded video from the list — it should be the only option available. The **Output Key Prefix** is for the subfolder we are using for the video. Leave this blank, since we have no subfolders in our bucket.

For **Output Details**, under **Preset**, select **Generic 480p 16:9**. For the **Output Key**, give your file a new name. We used *big_buck_bunny_linuxacademy_480p_h264.mp4*. Select *No* for **Create Thumbnails**. Leave the rest of the settings at default. **Create New Job**.

Navigate to **Jobs** through the left menu, select your *Pipeline ID* and click **Search**. You should see that your job is now progressing. Once finished, move to your output S3 bucket. Your transcoded video should now be available. Either through the **Actions** button or by right clicking, make the new video *public*.

To see if the video is viewable, return to your **CloudFront Dashboard** and retrieve your *Domain Name*.



	Delivery M	ID	Domain Name	Comment	Origin
□	Web	E10UTSQMJCDYAW	d36e5tgx9he9oa.cloudfront.net	-	lalab-output.
□	Web	E1G999NP7TSUOG	d2ol8fg7zg7svi.cloudfront.net	-	la-laboutput-
□	Web	E2I60NLYA5R0XR	d3s5f3fdugtaos.cloudfront.net	-	la4failover.s3
□	Web	E2XWHS0X8M4TC0	d2e3ayxdsd1kmv.cloudfront.net	-	dark-in-barn

Append the name of the new MP4 file after the URL. It should resemble:

```
d3oc4m1k4v8vne.cloudfront.net/big_buck_bunny_linuxacademy_480p_h264.mp4
```

Open your unique URL in you browser to view the video.

Streaming the Video to an HTML5 Page

To see the video work in an embedded situation, create an *index.html* file on your workstation and input the following code, replacing `d3oc4m1k4v8vne.cloudfront.net/big_buck_bunny_linuxacademy_480p_h264.mp4` with the URL for your video:

```
<video width="320" height="240" controls>
    <source src="d3oc4m1k4v8vne.cloudfront.net/big_buck_bunny_linuxacademy_480p_h264.mp4"
type="video/mp4">
</video>
```

Save and close. If you open this file in your web browser, you can view your streamed video.