videon

Configuring an HTTP Streaming Workflow with Videon's EdgeCaster, AWS MediaStore, and CloudFront

Project Name	AWS HTTP Streaming Workflow
Project Purpose	The purpose of this project is to validate and document the process of setting up and configuring a workflow for HTTP Streaming using AWS services as well as identify the expected results

Test Highlights

Purpose: Videon's EdgeCaster supports native HLS/DASH HTTP streaming. With features like multiple bit rate support and chunked CMAF support, users have the opportunity to create both low cost HTTP streaming and low latency HTTP streaming when using AWS MediaStore as origin server and AWS CloudFront as Content Delivery Network. Detailed methodology for setting up a Non-Ultra Low Latency, Ultra Low Latency, and Cross-device Synchronization workflow will be provided along with expected performance measurements.

Goal: To lay out a set of configuration instructions so that the intended audience can follow step-by-step instructions and have success in setting up an HTTP Streaming workflow using AWS MediaStore and CloudFront, observing similar performance results recorded in this document. Also, the results will be considered in terms of the benefits of using AWS services over other potential solutions.

Results: AWS MediaStore and CloudFront allow extremely easy setup and do not require user intervention after initial setup. Latency/stability performance is unaffected by the use of AWS services in an HTTP workflow, meaning AWS services were found to be highly reliable and efficient. All workflows (non-ULL, ULL, and Sync) were found to be supported in a workflow using AWS services.

Intended Audience: This workflow document assumes video streaming technical knowledge and the ability to set up or access an AWS account as well as have access to a player application that supports HTTP streaming for non-ULL, ULL, and/or Cross-device Synchronization (NexPlayer used in this document). With this knowledge, one should be able to use this information to successfully set up a workflow using AWS services within 45 minutes.

Purpose

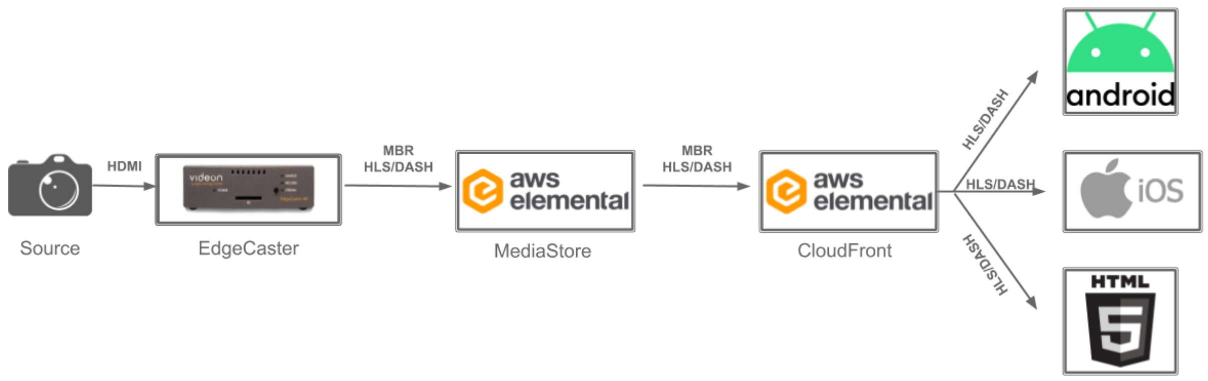
The purpose of this document is to record the details of what an HTTP Streaming workflow using AWS services requires, how to set it up, the possible technical use cases, and what the expected performance of each technical use case is.

Goal

The goal of this document is to produce a set of instructions where a Videon partner or user can reproduce the workflow using AWS services to support any of the possible technical use cases detailed in this document and to be able to determine success based on the expected performance also detailed in this document.

Block Diagram

This workflow and its variations can be represented by a single block diagram. What will change are the specific settings in various parts of the workflow. For example, turning Ultra Low Latency off for EdgeCaster and the players will create a workflow that is no longer low latency, but the key components of the block diagram will not have changed.



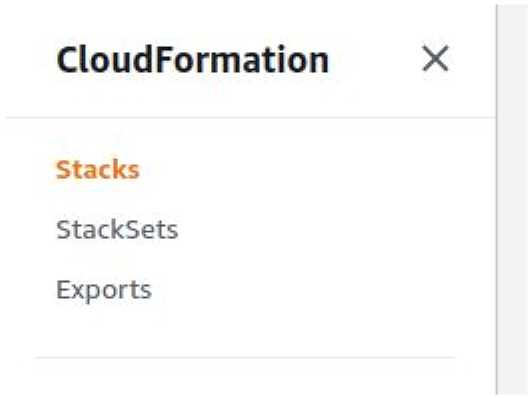
Instructions

To configure AWS **MediaStore** and **CloudFront**

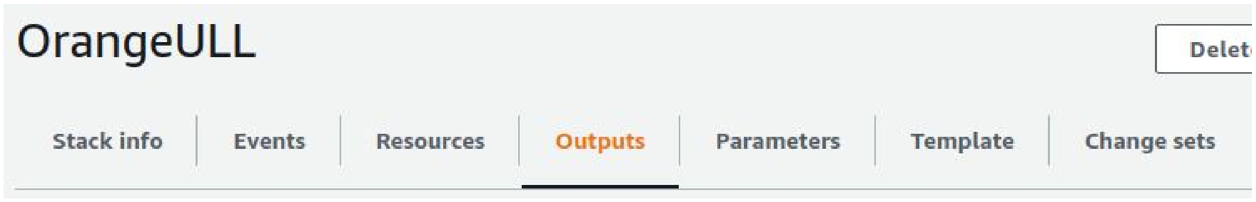
1. Open the CloudFormation template by clicking here: CloudFormation Template
2. Select the region closest to you from the title bar in the upper right corner of the CloudFormation screen, options are N. Virginia, Oregon, Seoul, Sydney, Tokyo, Frankfurt, Ireland, and Stockholm
3. From the “Create Stack” screen, select the “Next” button on the bottom right of the screen
4. From the “Specify stack details”, enter the following fields:
    a. **Stack Name** – The default is “ULL”, you can name it something different, but it cannot be the same name as an existing stack
    b. **MediaStoreContainerName** – The default is “MyContainer”, you can name is something different, but no spaces, just numbers/characters
    c. **EncoderIP** – This is the encoder's public IP
        i. If you want to allow traffic from anywhere, enter 0.0.0.0
    d. **EnableCloudFront** – The default is “False”. Set this to **true.**
5. Leave all the other fields unmodified, and select the “Next” button at the bottom right of the screen
6. From the “Configure stack options” screen select the “Next” button at the bottom right of the screen
7. From the “Review Stackname” screen, check the Capabilities acknowledgment and select the “Create stack” button at the bottom right of the screen.
8. When the stack is complete, you should see a green “CREATE_COMPUTE” check box under the Stacks column. When deploying with CloudFront set to true, this should take 20 minutes.

Take note of the MediaStore and CloudFront URLs for use when setting up EdgeCaster and viewer playback:

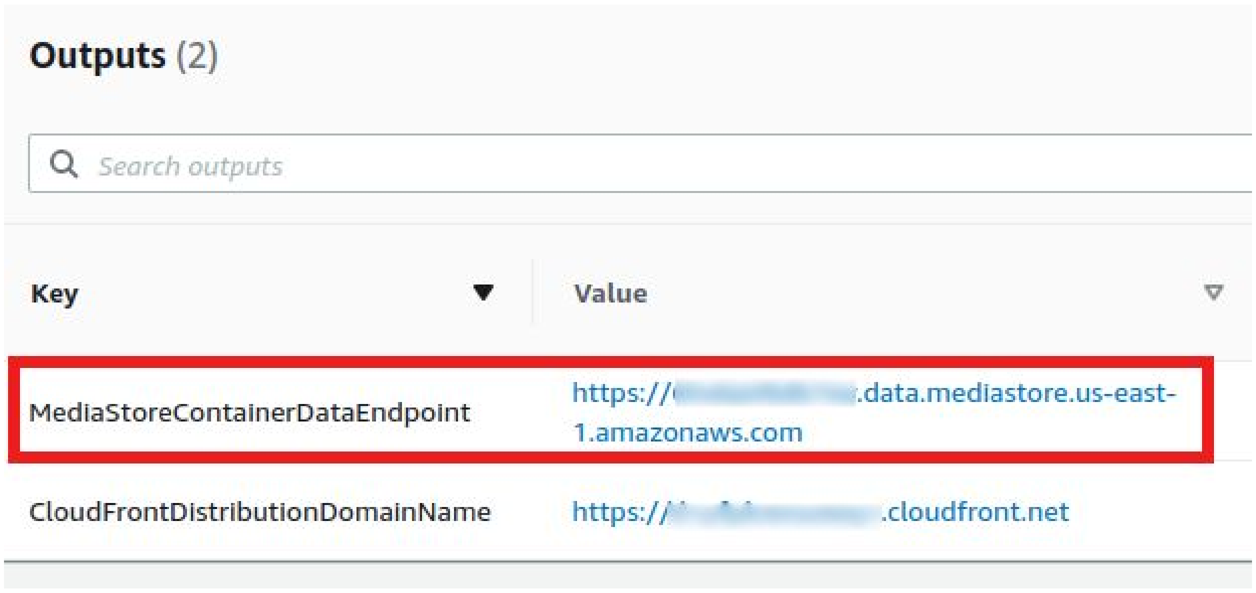
1. Navigate to AWS CloudFormation
2. Click on **Stacks** in the left sidebar



3. Click on the name of the stack created for this workflow
4. Click **Outputs**



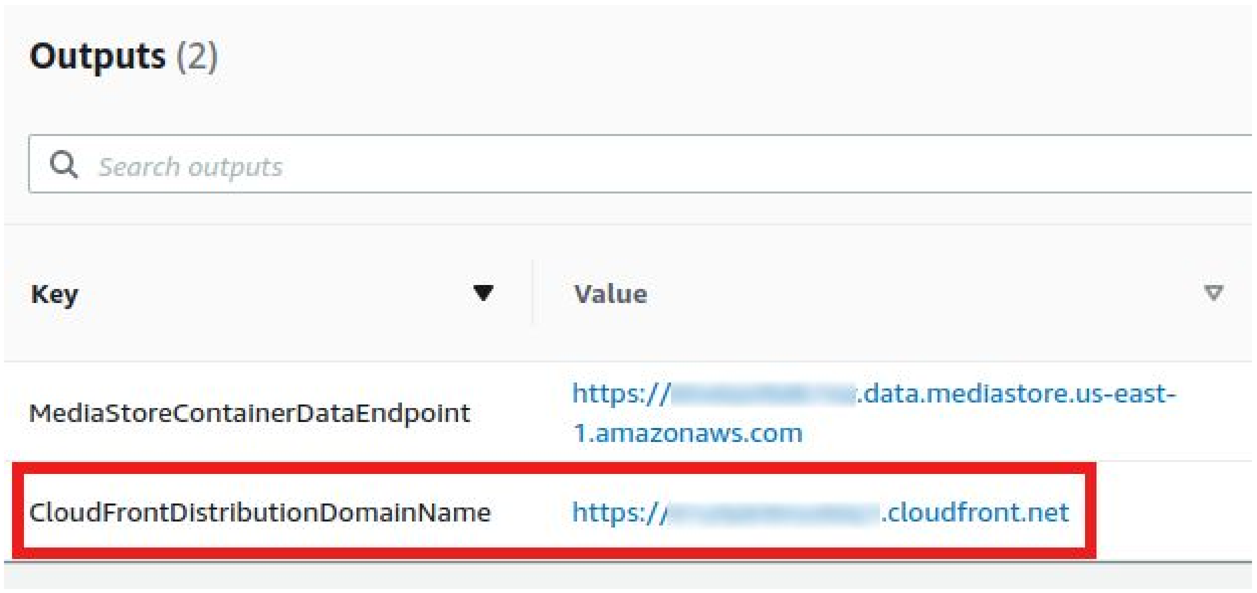
5. The **MediaStore URL** will be listed next to **MediaStoreContainerDataEndpoint**.
   a. This will be used to configure EdgeCaster
6. The **CloudFront URL** will be listed next to **CloudFrontDistributionDomainName**.
   a. This will be used to configure the player



Outputs (2)

Search outputs

Key	Value
MediaStoreContainerDataEndpoint	https://[redacted].data.mediastore.us-east-1.amazonaws.com
CloudFrontDistributionDomainName	https://[redacted].cloudfront.net

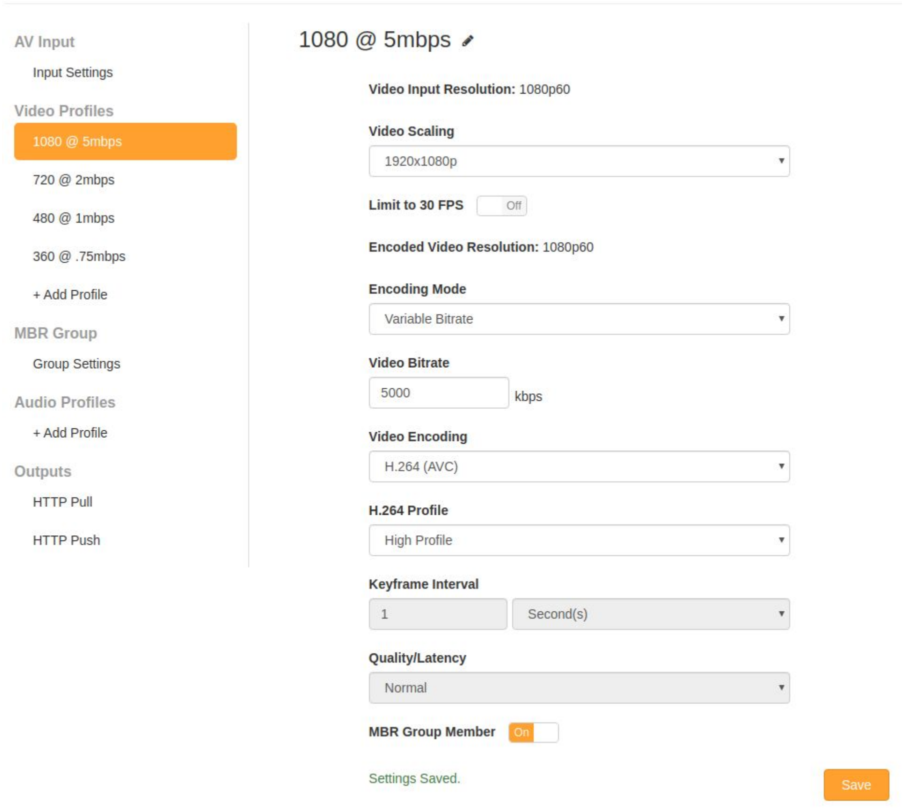


Outputs (2)

Key	Value
MediaStoreContainerDataEndpoint	https://[redacted].data.mediastore.us-east-1.amazonaws.com
CloudFrontDistributionDomainName	https://[redacted].cloudfront.net

To configure **EdgeCaster**

1. Click **+ Add MBR Group**
2. Click **Group Settings** (it appears in place of + Add MBR Group)
3. Set the respective settings to the desired configuration
   - The defaults here are good for a majority of use cases
   - If trying to get the absolute lowest latency possible, set **Quality/Latency** to **Lowest**
4. Click **Save**
5. Click **+ Add Video Profile** for the number of Video Profiles in the ladder
   - Do not exceed 4Kp30 for combined resolution/frame rate
   - For this example, the following ladder was used
     i. 1080p60 @ 5000kbps
     ii. 720p60 @ 2000kbps
     iii. 480p60 @ 1000kbps
     iv. 360p60 @ 750kbps
6. For each **Video Profile**, click the name in the left sidebar then
   - Click the name in the main configuration menu (should have a **pencil icon** next to it) and type the new name of the Video Profile
   - Set **Video Scaling** to the desired resolution for the profile
   - Configure the rest of the settings according to the desired ladder
   - Make sure to turn **MBR Group Member** to **ON**
   - Click **Save**



1080 @ 5mbps

Video Input Resolution: 1080p60

Video Scaling: 1920x1080p

Limit to 30 FPS: Off

Encoded Video Resolution: 1080p60

Encoding Mode: Variable Bitrate

Video Bitrate: 5000 kbps

Video Encoding: H.264 (AVC)

H.264 Profile: High Profile

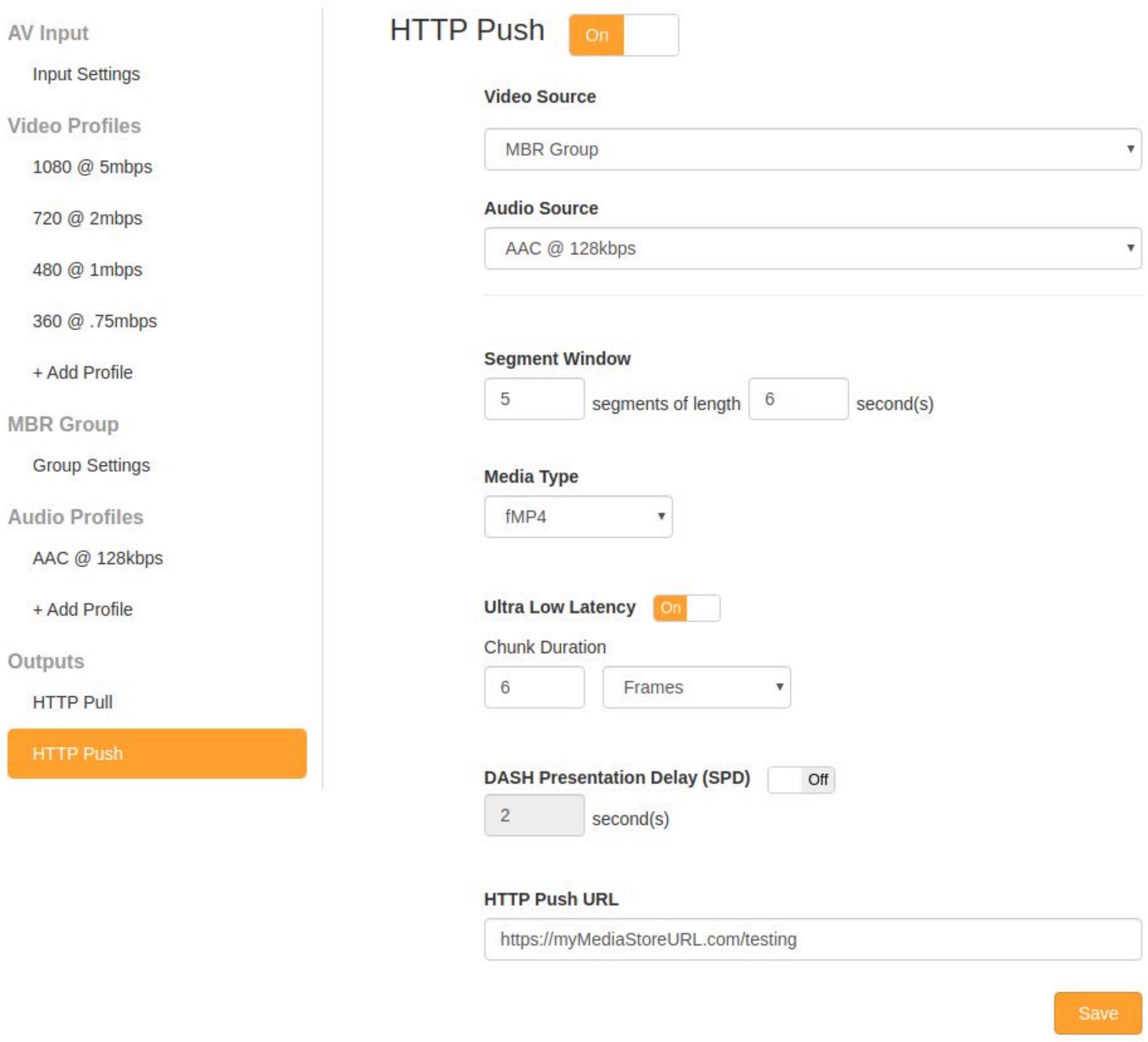
Keyframe Interval: 1 Second(s)

Quality/Latency: Normal

MBR Group Member: On

Settings Saved.

7. Under **Audio Profiles** click **+ Add Audio Profile**
8. Click the new **Audio Profile name** that appears
9. Click the name in the main configuration menu (should have a **pencil icon** next to it) and type the new name of the Audio Profile
10. Configure the desired settings
11. Click **Save**
12. In **HTTP Push**, select the **MBR Group** as the **Video Source**
13. Select the previously created Audio Profile and the **Audio Source**
14. Enter the **ingest URL** of the MediaStore container into the **HTTP Push URL** field
15. If setting up a low latency workflow, enable **Ultra Low Latency**
16. Turn **HTTP Push** ON and click **OK** or **Apply**
17. You can now view the stream at **[CloudFront URL]/master.m3u8 for HLS** or **[CloudFront URL]/manifest.mpd for DASH** (see player configuration below)


AV Input
Input Settings
Video Profiles
1080 @ 5mbps
720 @ 2mbps
480 @ 1mbps
360 @ .75mbps
+ Add Profile
MBR Group
Group Settings
Audio Profiles
AAC @ 128kbps
+ Add Profile
Outputs
HTTP Pull
HTTP Push

HTTP Push On

Video Source: MBR Group

Audio Source: AAC @ 128kbps

Segment Window: 5 segments of length 6 second(s)

Media Type: fMP4

Ultra Low Latency: On

Chunk Duration: 6 Frames

DASH Presentation Delay (SPD): Off
2 second(s)

HTTP Push URL: https://myMediaStoreURL.com/testing

Save

To configure the **player**

This document is intended to demonstrate the use of AWS services in an HTTP Streaming workflow with EdgeCaster. Different players have their own Videon-created documents detailing the tests run with each individual player. For more information on those documents, please contact Videon.

For this document, the NexPlayer Android player will be used as an example as it supports HLS and DASH, in ULL and non-ULL workflows as well as cross-device synchronization, allowing the most variations of HTTP Streaming workflows to be demonstrated with a single set to configuration instructions.

1. From the main screen, tap the three vertical dots in the top right corner and select settings

2. Configure the settings according to the following
    a. Low Latency (**ignore if non-ULL workflow**)
        i. Enabled = True
        ii. Low Latency Buffer Option = Auto Buffer Management
    b. SPD Settings (**ignore if not a cross-device synchronization workflow**)
        i. Enabled = True
        ii. SPD delay time = 3000ms
        iii. SPD sync value = 500ms
        iv. SPR too much sync diff value = 10000ms
3. Navigate back to the main screen
4. Tap streaming at the top
5. Tap **GO TO URL**
6. In the **URL** field, enter the CloudFront URL from the end of the AWS instructions above
7. Tap **PLAY** at the bottom to start the stream

Results and Observations

Summary

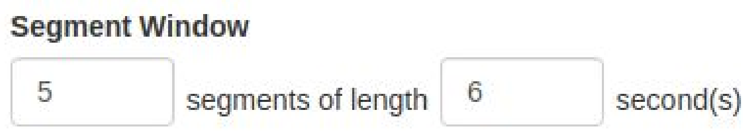
The below table summarizes the expected results, by workflow, with the above settings

Workflow	Expected result
Non-ULL, no Sync	6-30 second latency
ULL, no Sync	<3 second latency
Non-ULL + Sync	**DASH**: 7.5-8.5 second latency across devices **HLS:** 8.5-9.5 second latency across devices
ULL + Sync	**DASH**: 1.5-2.5 second latency across devices **HLS:** 2.5-3.5 second latency across devices

Non-ULL

When streaming with Ultra Low Latency turned OFF on EdgeCaster and Low Latency disabled on the player, the expected workflow will be non-ULL. This is the base workflow for HTTP Streaming as it is the most readily used since ULL workflows are just beginning to see more adoption.

The expected results for a non-ULL workflow will be that the video playback has a delay of at least the **Segment Length**, but up to the entire **Segment Window** or more. Segment Length can be found in the HTTP Push setting for EdgeCaster as part of the Segment Window:


Segment Window
5 segments of length 6 second(s)

In this case, the minimum latency would be 6 seconds, but the latency will typically be up to 30 seconds (number of segments x Segment Length) or longer. In a non-ULL workflow, the video is downloaded by the player in segments (i.e. for this case, every 6 seconds).

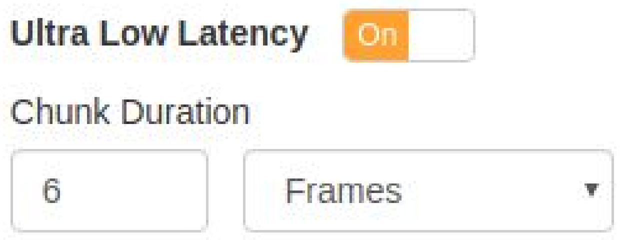
Latency tends to be higher when playing back HLS rather than DASH. This is because most HLS players keep a buffer of multiple segments, where many DASH players start playing as soon as a full segment is downloaded.

A “low latency” non-ULL workflow can be simulated by reducing the Segment Length and Number of Segments, but is not recommended as it can cause playback stability issues. Instead, using the Ultra Low Latency features on EdgeCaster and the player are recommended as they are designed for ULL applications.

ULL

When streaming with Ultra Low Latency turned ON on EdgeCaster and Low Latency enabled on the player, the expected workflow will be ULL. ULL workflows are beginning to see more adoption as the entire pipeline is now supported (encoder, cloud, and player), as is detailed by this document.

The expected results of this workflow will be that the video playback has a delay of at least the **Chunk Duration** (plus up to a few hundred milliseconds for internet transfer time), but up to the **Segment Length**. Segment Length is identified in the [section above](#). Chunk Duration can be found in the HTTP Push settings on EdgeCaster:


Ultra Low Latency On
Chunk Duration
6 Frames

In this case, the minimum latency would be 6 frames (again, plus up to a few hundred milliseconds). In a ULL workflow, the video is downloaded in chunks, which make up segments. In this case, this would mean playback could begin with a buffer as little as 100ms. Typically, the player buffers at least 1 second to ensure higher levels of stability. Adding together the various latencies from parts of the workflow, the glass-to-glass latency for a stable playback experience can be as little as 1.4 seconds, but is typically slightly higher as production workflows use less aggressive buffers to ensure stable, high-quality experiences.

Latency tends to be higher when playing back HLS rather than DASH. This is due to DASH being designed for chunk-based data transfer, where community implementations of HLS playback have introduced modifications to support chunk-based data transfer since ULL is not officially supported by HLS. With the introduction of Apple's LL-HLS, this may change.

In the example provided with EdgeCaster, MediaStore, CloudFront, and NexPlayer, stable, high-quality, ULL playback can achieve latencies as low as 2.3 seconds.

Cross-device Synchronization

With **DASH Presentation Delay (SPD)** ON on EdgeCaster and **SPD Settings** enabled on the player, the expected workflow will include Cross-device Synchronization. Cross-device Synchronization can be included for both ULL and non-ULL workflows.
The expected results for a Cross-device Synchronization workflow will be that video playback on multiple devices will be synchronized within 1 second. The window of synchronization can be lower using more aggressive settings, but to have a balance of stability, latency, and synchronization, synchronization within 1 second is the target for this workflow.

Including Cross-device Synchronization will add additional latency based on the respective **Suggested Presentation Delay (SPD)** setting. For HLS, the SPD setting is determined by the player as HLS is not designed to pass an explicit SPD setting in the manifest file form the encoder. For DASH, the SPD setting is determined by the encoder (in the DASH manifest file).

DASH Presentation Delay (SPD) was set to 2 seconds on EdgeCaster, meaning playback should be expected to synchronize in a 1-second window 2 seconds from the presentation time. In a non-ULL workflow (presentation time/latency of 6 seconds), that means multiple devices should be synchronized at 7.5-8.5 seconds of glass-to-glass latency. In a ULL workflow (latency of 2.3 seconds) this number is different as the observed latency is as low as possible rather than representative of the presentation time. The presentation time in a ULL workflow is the start time of the stream (0 second latency), that means devices should be synchronized at 1.5-2.5 seconds of playback latency in a ULL DASH workflow with the above settings.

The **SPD delay time** on Nexplayer was set to 3000ms with a **SPD sync value** of 500ms. Taking into consideration the same details from the previous paragraph, a non-ULL workflow should see Cross-device Synchronization at 8.5-9.5 seconds of latency and a ULL workflow should see Cross-device Synchronization at 2.5-3.5 seconds of latency.

Conclusion

The main conclusion of the testing performed to document MediaStore and CloudFront for HTTP Streaming content distribution is that the only effort related to AWS required for this workflow is the initial setup of the AWS services. AWS MediaStore and CloudFront serve as a tool to allow the scalability of HLS and DASH streaming regardless of specific settings on the encoder and player.

MediaStore serves as an origin server, holding the HLS/DASH streams in a centralized cloud location to allow CloudFront, as the Content Delivery Network, to reference the stream files for delivery to hundreds of thousands of concurrent devices.

The main benefit of using AWS services for HTTP Streaming is that the setup and configuration can be done easily via the referenced CloudFormation template. Once the MediaStore and CloudFront instances are set up using that CloudFormation template, they require no further interaction.

While AWS offers an extremely broad selection of services, they also offer the tools to easily set up an HTTP Streaming workflow using MediaStore and CloudFront. Once setup is complete, the user can treat MediaStore and CloudFront as set-and-forget components to the workflow.