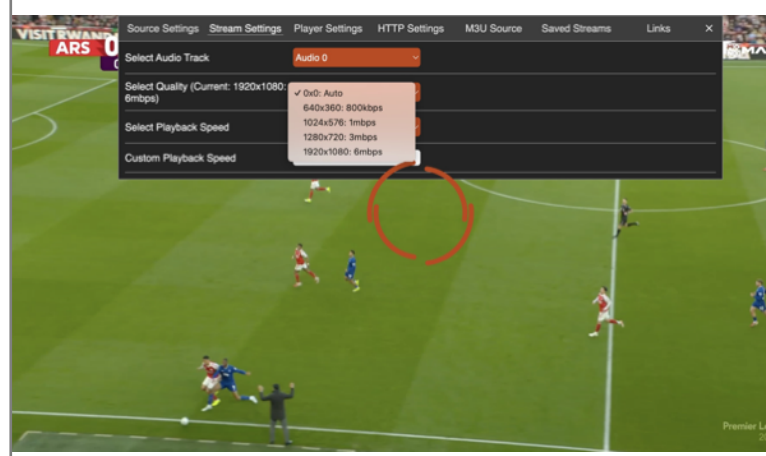
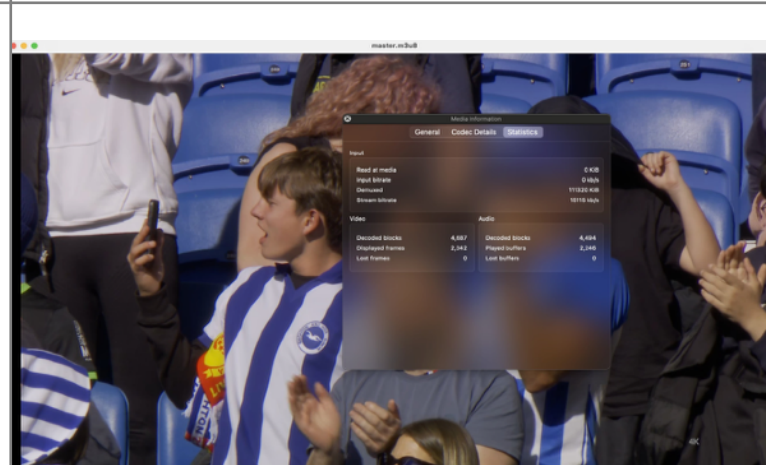
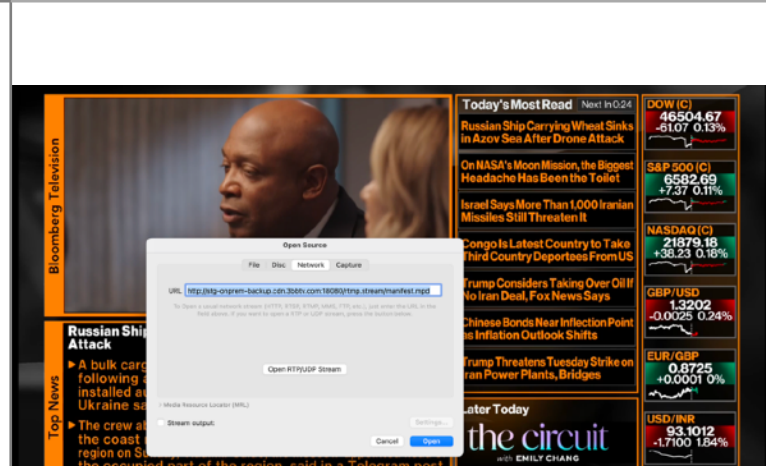
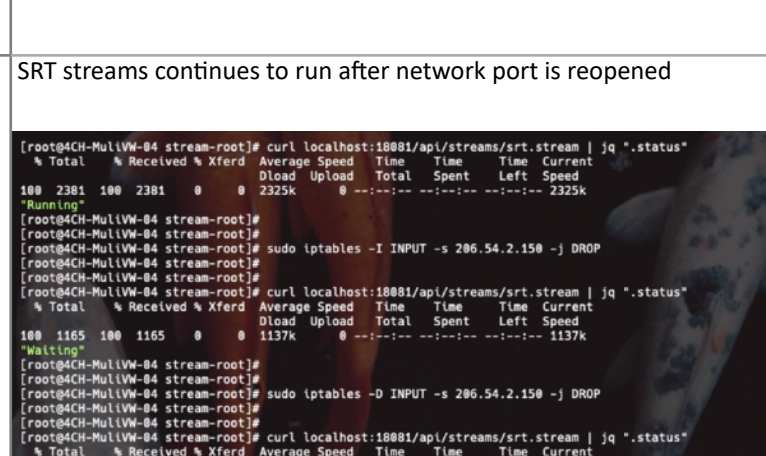
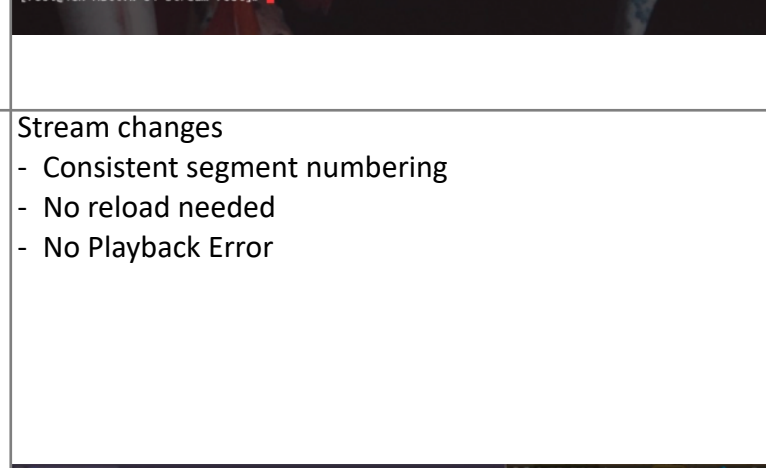
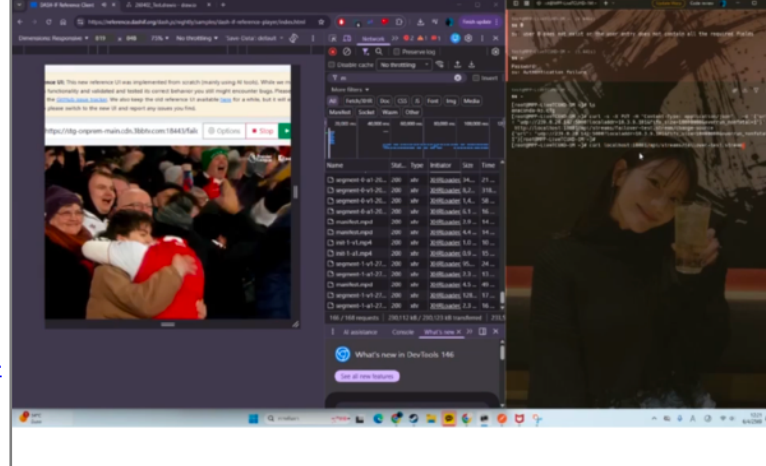
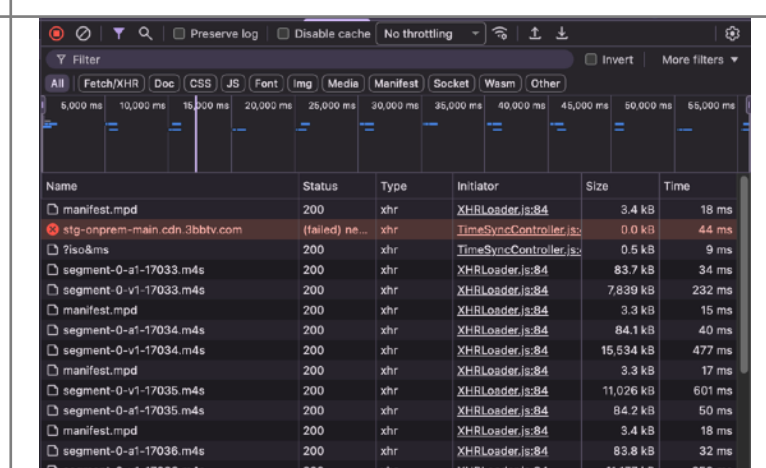
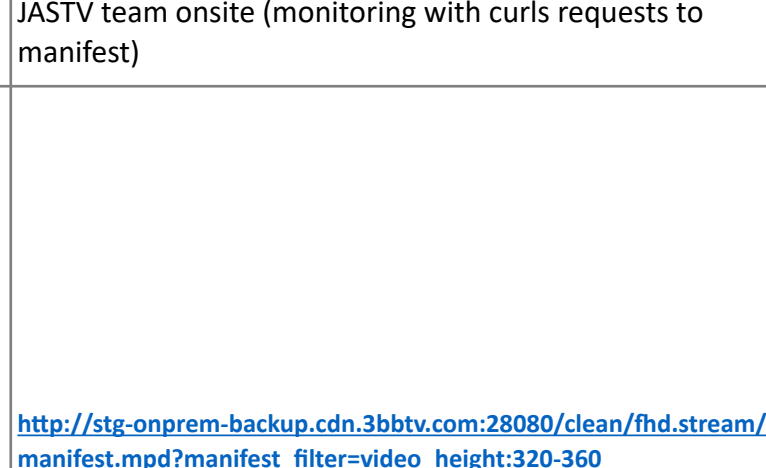
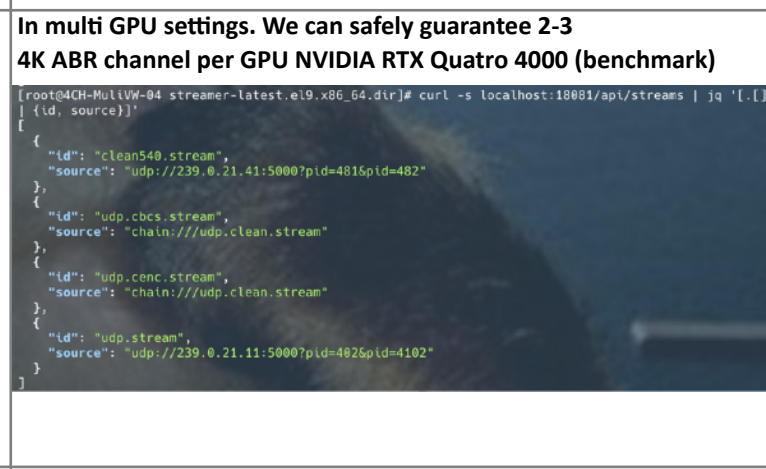
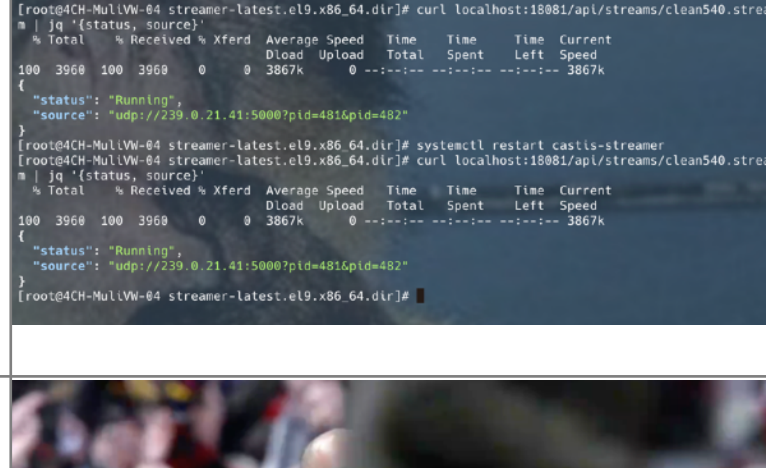
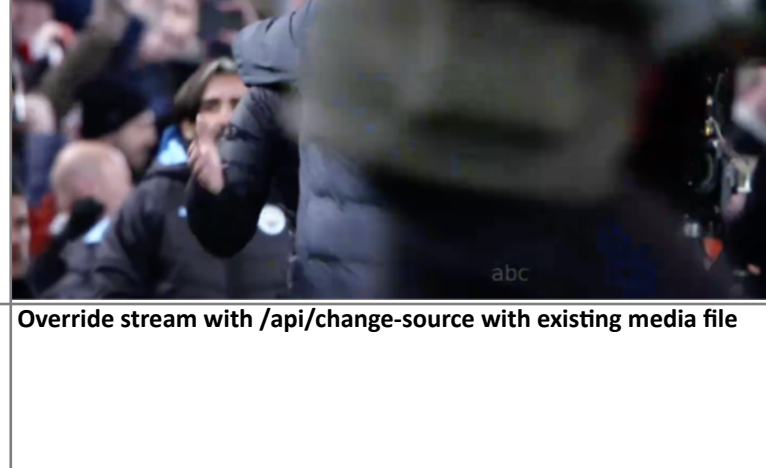
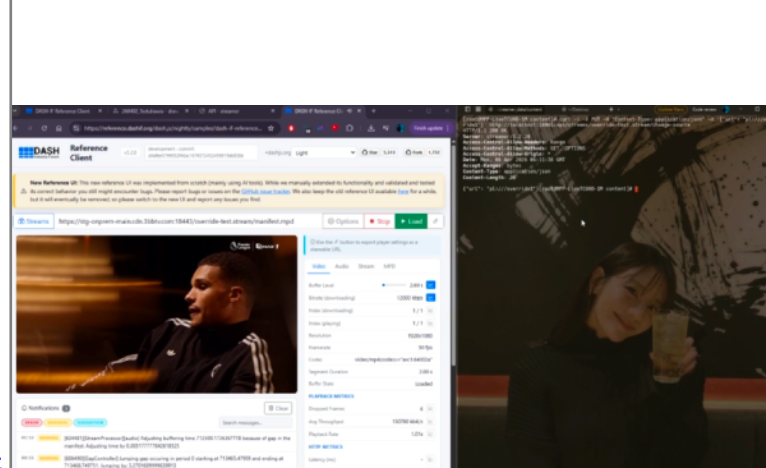
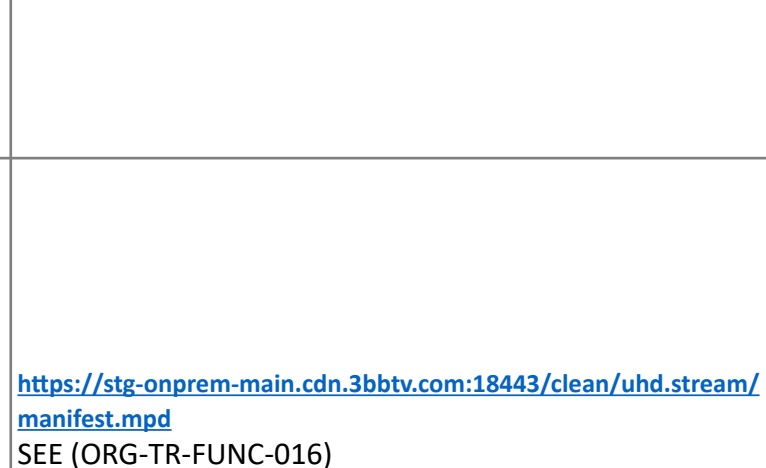
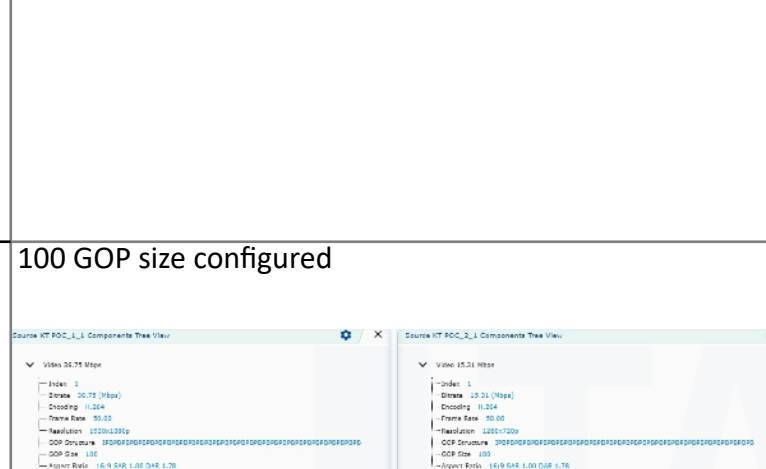
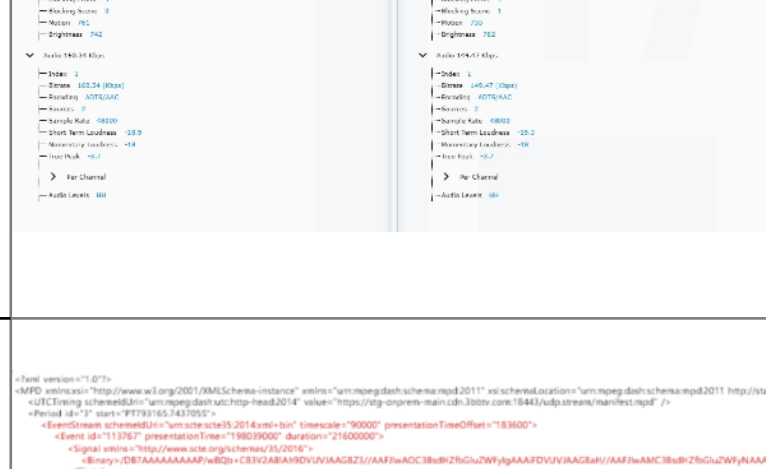


[illegible]

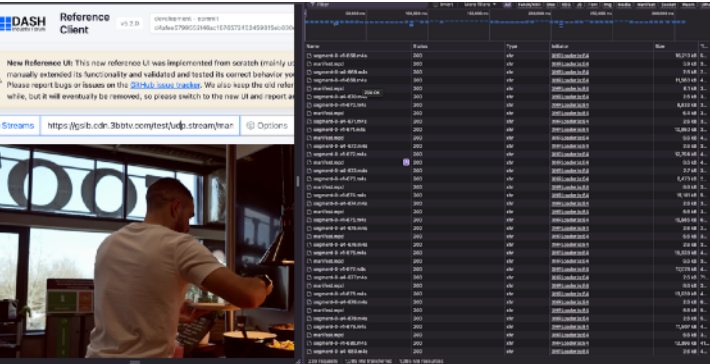
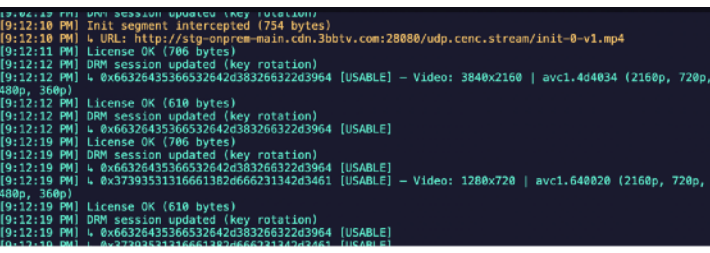
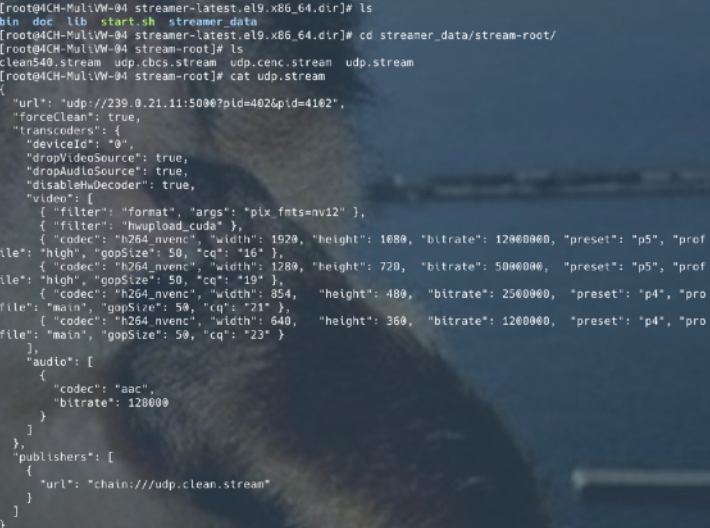
List of Function Test						
#	Category	Part No.	Test Category	TC ID	Test Item	TC Result
1	Orign Streamer / Transcoer	1	Live Ingest	ORG-TR-FUNC-001	UDP Ingest	
				ORG-TR-FUNC-002	SRT Ingest	
				ORG-TR-FUNC-003	RTMP Ingest	
				ORG-TR-FUNC-004	Ingest reconnect under loss	
				ORG-TR-FUNC-005	Input Redundancy	
		2	CMAF	ORG-TR-FUNC-006	cmaf init segement validity	
				ORG-TR-FUNC-007	Segement duration stability	
		3	Manifest	ORG-TR-FUNC-008	Manifest creation timing	
				ORG-TR-FUNC-009	Manifest stability	
				ORG-TR-FUNC-010	Manifest Filtering	
		4	Channel Maangement Service	ORG-TR-FUNC-011	multi-channel service	
				ORG-TR-FUNC-012	Origin Restart (no DRM)	
				ORG-TR-FUNC-013	Crawling Text	
				ORG-TR-FUNC-014	Stream override	
		5	Transcoder	ORG-TR-FUNC-015	Multi Profie Encoding & ABR	
				ORG-TR-FUNC-016	GOP Alignment	
				ORG-TR-FUNC-017	SCTE-35 Sginal generation	
				ORG-TR-FUNC-018	Multi layer logo insertion	
				ORG-TR-FUNC-019	Audio level adjustment	
2	Encryption	6	CBCS/CENC Encryption	ENC-FUNC-001	CMAF init segment validity	
				ENC-FUNC-002	CBCS/CENC encryption	
				ENC-FUNC-003	Clear Key playback	
				ENC-FUNC-004	SPEKE v2 key retrieval	
		7	DRM	DRM-FUNC-001	License playback	
				DRM-FUNC-002	Unauthorized playback	
				DRM-FUNC-003	Key rotation	
				DRM-FUNC-004	DRM continuity on origin restart	
3	Origin Streamer HA	8	HA	ORIGIN-HA-001	Active/Active origin	
				ORIGIN-HA-002	Init segement stability	
		9	Crypto	ORIGIN-HA-003	Crypto state continuity	
		10	Service	ORIGIN-HA-004	Manifest persistence	
				ORIGIN-HA-005	Channel restart	
4	End to End HA	11	Live Origin Streamer	E2E-HA-001	Single node failure	
				E2E-HA-002	Node Recovery	
				E2E-HA-003	Site Failure	
				E2E-HA-004	Origin restart mid-sgement	
				E2E-HA-005	Origin Shield failover validation	
		12	Crypto	E2E-HA-006	DRM continuity on origin failover	
				E2E-HA-007	Key rotation during failover	
		13	Shield	E2E-HA-008	Origin Shield node failover	
				E2E-HA-009	Origin Shield site failover	
		14	Multi-Site	E2E-HA-010	Full site A blackout	
				E2E-HA-011	Site recovery	
		15	Monitoring	E2E-HA-012	Alert validation	
				E2E-HA-013	Failover event visibility	
		16	Origin Streamer Aging Test	E2E-HA-014	72-hour streaming stability	
				E2E-HA-015	Origin streamer CPU stability	
				E2E-HA-016	Origin streamer memory stability	

ORIGIN STREAMER TEST CASES

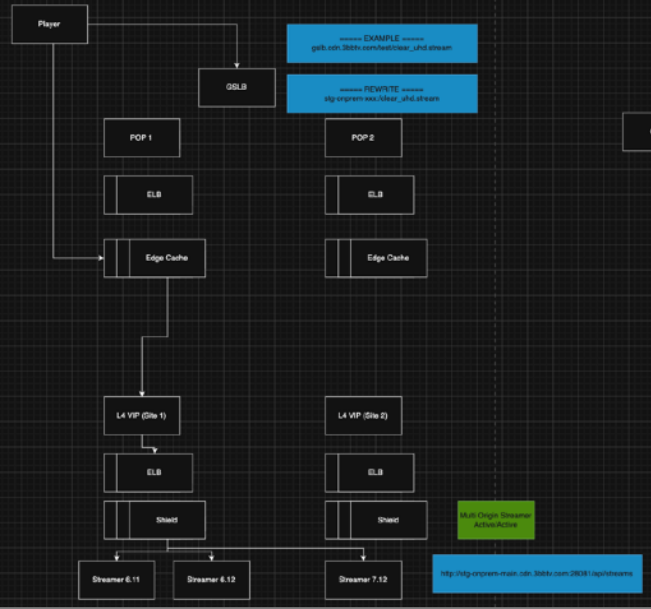
Project Name		CDN Update for Monomax Service		Tester		Test date		
Test Category		ORIGIN STREAMER / TRANSCODER		SCOPE	Origin Streamer AND Transcoder functional validation for live ingest, CMAF packaging, manifest behavior, fallback streaming, and service recovery.			
Test Environment		Live Ingest generated on a PC using OBS Studio and FFmpeg, delivered to Live Origin Streamer via UDP, SRT, and RTMP. Content is packaged as clear CMAF and served through CDN for playback validation.			6.12 -> UDP [11] SRT 7.12 -> DRM vs Clean			
PART #	CATEGORY	TEST ITEM		CONFIGURATION	HOW TO TEST	EXPECTED RESULT	TC RESULT	
1	Live Ingest	ORG-TR-FUNC-001	UDP Ingest	• UDP MPEG-TS source (IGMPv2, v3) • CBR TS, PAU/PMT valid • Origin Streamer UDP listen port configure	1) Send UDP TS using ffmpeg or hardware encoder 2) Start origin channel 3) Request manifest and play via player	• Stream starts • No continuity counter error • Stable playback		
		ORG-TR-FUNC-002	SRT Ingest	• SRT listener mode • Caller/Listener configured • Latency configured (e.g. 200ms)	1) Start SRT sender 2) Verify origin receives input 3) Play via CDN/player	• Input lock achieved • Stable playback • No packet loss visible		
		ORIGIN-FUNC-003	RTMP Ingest	• RTMP push endpoint enabled • GOP aligned encoder	1) Push RTMP stream 2) Start channel 3) Request manifest	• RTMP accepted • Stream plays normally		
		ORG-TR-FUNC-004	Ingest reconnect under loss	• SRT or RTMP ingest • Reconnect enabled	1) Inject packet loss/jitter (tc/netem) 2) Stop input briefly 3) Restore input	• Origin auto-reconnects • Channel continues without manual reset		
		ORG-TR-FUNC-005	Input Redundancy & Auto-Failover (Primary/Backup)	• Primary + Backup input configured • Two identical streams available • Auto-switch enabled	1) Start both ingest pipelines 2) Verify both TS streams via ffprobe 3) Confirm transcoder locks to Primary 4) Stop Primary input 5) Observe failover behavior 6) Restore Primary input 7) Observe recovery behavior	• Both inputs decoded correctly • No instability when both present • Automatic switch to Backup on Primary loss • No channel restart required • No crash / freeze / long buffering • If designed: Automatic switch-back or stable manual recovery API for stream switching (https://fastlydev.github.io/streamer/docs/resource/api/#P4E8YSD0JBCNECN9D5B4XEBNB8Y8C-NECN8A9AAKEDPSBAIB8BN8BHA6NIBC-%E8%82%B4NECN9D9B8-NECN8B8BCNECN8AIAA-NECNADPSBB8NECC8B2%84)	 	
2	CMAF	ORIGIN-FUNC-006	cmaf init segment validity	• CMAF output enabled • FMP4 packaging	1) Request init segment directly 2) Inspect with mpdump/ffprobe	• Valid FMP4 header • Track IDs correct • No decoder error		
		ORG-TR-FUNC-007	Segment duration stability	• Target segment duration (e.g. 2s)	1) Capture 20+ consecutive segments 2) Parse manifest (MPD/m3u8) for declared durations 3) Extract actual durations using mediainfo/mp4box 4) Calculate variance, drift, and consistency	• Duration within tolerance • No drift or missing segments	More than 5 Min consecutive playback verified with JASTV onsite (monitoring with playback quality)	
3	Manifest	ORG-TR-FUNC-008	Manifest creation timing	• Channel auto-start enabled	1) Start channel 2) Measure time until manifest available	• Manifest available		
		ORIGIN-FUNC-009	Manifest stability	• Live manifest sliding window	1) Continuously request manifest 2) Observe updates over time	• No invalid tags • No discontinuity unless expected	More than 5 Min consecutive playback verified with JASTV team onsite (monitoring with curls requests to manifest)	
		ORG-TR-FUNC-010	Manifest Filtering via Query String	• Full ABR ladder configured (SD / HD / UHD) • Middleware assigns service tier (Basic / Standard / Premium) • Player appends manifest filtering parameter in query string • CDN configured to forward manifest filtering query string and include in cache key • Origin filtering enabled based on query string parameter • Cache key normalized for manifest_filter	1) Request manifest with query string for Basic tier ?manifest_filter=video_height:240-480 2) Verify only SD profiles returned 3) Request with Standard tier ?manifest_filter=video_height:240-720 4) Verify SD + HD profiles returned 5) Request with Premium tier (no filter or full ladder range) ?manifest_filter=video_height:240-2160 or no parameter 6) Verify full ABR ladder returned 7) Request without query parameter 8) Verify default tier behavior (define expected default ladder) 9) Send invalid filter value ?manifest_filter=invalid_value 10) Verify proper error handling or fallback behavior 11) Repeat requests to validate cache behavior (confirm cache HIT and no cross-tier cache bleed) 12) If dual parameters supported, test with: aws.manifestfilter=video_height:240-720&manifest_filter=video_height:240-720 13) Verify parameter precedence and cache separation behavior	• Only authorized renditions visible per tier • No restricted profiles exposed • Manifest syntax remains valid • Playback successful for allowed profiles • Default policy applied if header missing (per design) • Invalid tier handled correctly (deny or fallback) • No cross-tier cache leakage • No manifest corruption under concurrent requests http://stg-onprem-backup.cdn.3bbtv.com:28080/clean/fhd.stream/manifest.mpd?manifest_filter=video_height:320-360 http://stg-onprem-backup.cdn.3bbtv.com:28080/clean/fhd.stream/manifest.mpd?manifest_filter=video_height:1050-1200		
4	Channel Management Service	ORG-TR-FUNC-011	multi-channel service & Channel Lifecycle Management	• Multiple channels configured (UDP/SRT/RTMP/HLS/DASH mix) • Different profiles assigned (HD/UHD/HDR/SDR) • System resource monitoring enabled	1) Create multiple channels 2) Modify all renditions/channel configuration 3) Start multiple channels simultaneously 4) Stop one channel while others running 5) Delete one channel 6) Monitor CPU/memory during operations	• All channels start successfully • No cross-channel impact • Channel modification applies correctly • Stopping one channel does not affect others • Deleting one channel releases resources properly • System remains stable under load		
		ORIGIN-FUNC-012	Origin Restart (no DRM)	• Stateless mode • Channel config persisted	1) Restart origin process/node 2) Request manifest again	• Channel recovers automatically • No full reconfiguration needed		
		ORG-TR-FUNC-013	Crawling Text Service & Lifecycle Management	• Multiple channels running • Crawling text content configured • Speed, position, font style configured • Schedule defined (immediate + timed)	1) Create crawling text task for multiple channels 2) Start channels and verify text overlay 3) Update text content while running 4) Modify schedule time 5) Stop crawling text on one channel 6) Delete crawling task 7) Monitor CPU/memory during operations	• Crawling text appears correctly on assigned channels • Updates applied without stream interruption • Schedule triggers at correct time • No cross-channel impact • Cue-duration matches scheduled duration • Deleting task removes overlay immediately • No cross-channel impact • System remains stable under load		
		ORG-TR-FUNC-014	Stream Override & Priority Management (Fallback + Emergency)	• Primary ingest configured • Backup promotional slate configured • Emergency asset registered • Priority rule defined: Emergency > Fallback > normal	1) Start channel with primary ingest 2) Stop primary ingest 3) Verify fallback slate activates 4) Trigger emergency during fallback 5) Cancel emergency 6) Restore primary ingest	• Seamless switch to fallback when ingest fails • Emergency overrides fallback immediately • After emergency ends, system returns to fallback if ingest still down • When ingest is restored, stream returns to normal • Playback continues without interruption API for stream switching (https://fastlydev.github.io/streamer/docs/resource/api/#P4E8YSD0JBCNECN9D5B4XEBNB8Y8C-NECN8A9AAKEDPSBAIB8BN8BHA6NIBC-%E8%82%B4NECN9D9B8-NECN8B8BCNECN8AIAA-NECNADPSBB8NECC8B2%84)		
5	Transcoding	ORIGIN-FUNC-015	Multi-Profile Encoding & ABR Ladder Management	• Multiple output profiles configured (SD/HD/UHD) • Fixed ABR ladder defined • GOP alignment enabled • Mixed frame rate inputs (25/50 fps)	1) Start transcoding job with ABR ladder 2) Modify all renditions generated 3) Capture output TS and verify SCTE-35 PID via analyzer 4) Request HLS/DASH manifest during cue period 5) Monitor CPU/GPU usage	• All renditions generated correctly • GOP aligned across all profiles • Seamless bitrate switching • No frame drops or sync issues • Resource usage stable under load	https://stg-onprem-main.cdn.3bbtv.com:18443/clean/uhd.stream/manifest.mpd SEE (ORG-TR-FUNC-016)	
		ORG-TR-FUNC-016	GOP alignment	• Fixed GOP size • Aligned across renditions	1) Analyze output segments 2) Compare keyframe timing	• GOPs aligned across all profiles • Suitable for CMAF packaging		
		ORG-TR-FUNC-017	SCTE-35 Event Generation & End-to-End Signaling	• Origin SCTE Scheduler API enabled • Transcoder supports SCTE-35 insertion in MPEG-TS • HLS/DASH packaging enabled (CMAF optional) • Test channel running (live input)	1) Start live channel 2) Use Origin API to schedule SCTE-35 event (cue-out with defined duration) 3) Capture output TS and verify SCTE-35 PID via analyzer 4) Request HLS/DASH manifest during cue period 5) Verify correct marker insertion (e.g., DATERANGE / EventStream) 6) Observe return-to-program after cue duration 7) Repeat under multiple concurrent channels	• SCTE-35 packet correctly inserted in TS with accurate PTS • Cue timing aligned with segment boundary • Manifest contains correct signaling (no malformed tags) • Cue-duration matches scheduled duration • Seamless playback (no freeze / AV sync issue) • Return-to-program executed correctly • No cross-channel impact • System stable under load		
		ORIGIN-FUNC-018	Multi-Layer Logo & Overlay Rendering	• Up to 4 logo layers configured • PNG/JPG/GIF assets • Transparency enabled • Position & priority defined	1) Start transcoding with multiple logos 2) Verify layering order 3) Update one logo dynamically 4) Remove one layer during runtime 5) Monitor resource usage	• Logos rendered in correct order • Transparency preserved • Dynamic update applied without restart • Removing one layer does not affect others • No performance degradation		
		ORG-TR-FUNC-019	Audio Encoding & Level Adjustment	• AAC and Dolby E configured • LUFS monitoring enabled • Independent Main/Backup level control	1) Start transcoding job 2) Adjust audio level for Main 3) Adjust audio level for Backup 4) Measure LUFS output	• Audio encoding correct format • Level changes applied immediately • LUFS values within expected range • No clipping or distortion	// Loudness config for standardizing volumes across streams(channels) "transcoders": { "audio": { "filter": "loudnorm", "args": "1=-17" }, "codecs": "aac", "bitrate": "128000" } }	

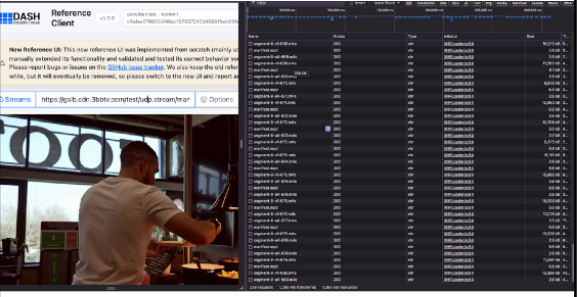
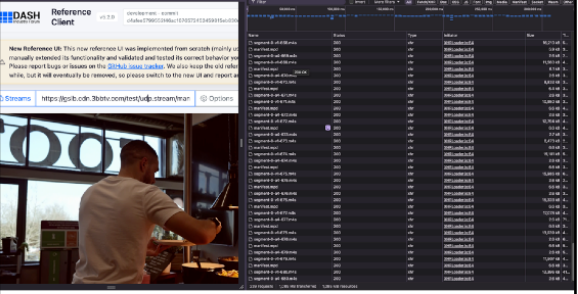
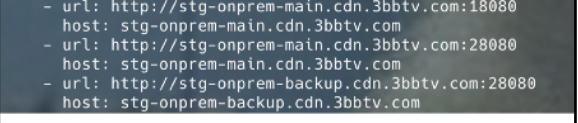
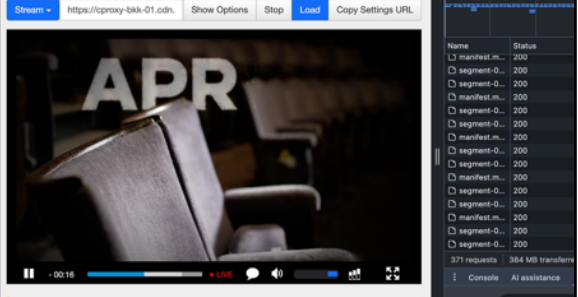
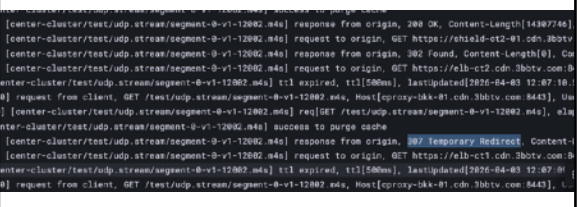
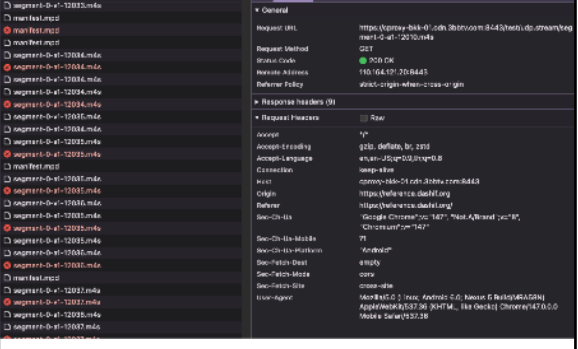
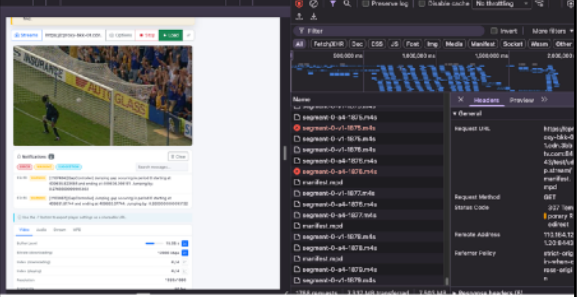
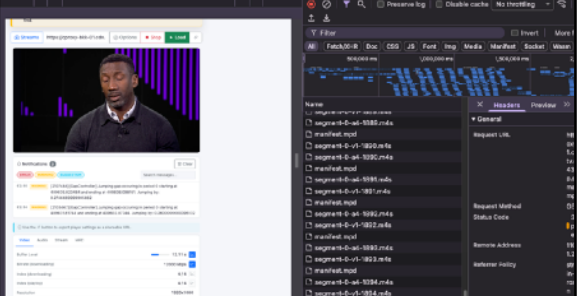
ORIGIN STREAMER HA

Project Name	CDN Update for Monomax Service	Tester		Test date		
Test Category	ORIGIN STREAMER HA	SCOPE	- Validate Origin Streamer high availability and failover behavior - Verify seamless or minimal-disruption failover for live channels - Confirm segment, manifest, and crypto state continuity			
Test Environment	The test environment includes active and standby origin streamers with shared object storage for segment and state persistence. Live ingest is delivered via SRT, with the standby origin in warm-ready mode. Testing focuses on origin-level high availability and failover behavior					

Part #	Category	Test Item		Configuration	How to Test	Expected Result	TC Result
8	HA	ORIGIN-HA-001	Active/Active origin	- Two origin streamer nodes (Active/Active) - Same channel configuration on both origins - Cache proxy (cproxy) handles failover logic - Active-first routing strategy at cache layer	- Start live channel on both origin - Verify segment generation and playback - Kill one origin process	- Channel resumes without reconfiguration - Continuous playback without restart, with short rebuffer	E2E CDN were setup with Cache Server having 3 Origin Nodes. Playback on GSLB and Proxy -> No errors were found - No quality of stream were compromised (Verified with onsite playback with JASTV team) 
		ORIGIN-HA-002	Init segment stability	- CMAF packaging enabled - No forced channel restart on failover	- Start live channel - Capture init segment before failover - Trigger origin failover - Capture init segment after failover	- Init segment remains compatible - No codec or track mismatch - Existing players continue playback	Continuous segments with consistent numbering were captured while streaming, even with node restarts 
9	Crypto	ORIGIN-HA-003	Crypto state continuity	- CENC enabled (ClearKey or DRM) - Same crypto configuration on both origins - Stateless or synchronized crypto handling	- Start encrypted live channel - Begin playback - Trigger origin failover - Observe playback	- Encryption continues seamlessly - No decryption error	Continuous Key rotation after we shut down an origin node. (Verified with onsite playback with JASTV team) 
10	Service	ORIGIN-HA-004	Manifest persistence	- Manifest persisted or deterministically regenerated - Channel metadata stored outside process memory	- Start live channel - Continuously request manifest - Trigger origin failover	- Manifest always available - No 404 or empty response - Timeline remains monotonic	Config stored in directory and not memory. (Manifest consistency is confirmed with HA and E2E testing Node shutdowns and recovery, no manifest error) 
		ORIGIN-HA-005	Channel restart	- Channel configuration persisted - Transcoder remains running	- Start live channel - Restart origin process - Observe channel recovery	- Channel restarts automatically - No manual re-creation - Playback recovers	.stream config is stored in streaming server's directory as files on disk, where its read by streamer. Channels are always restarted based on existing .stream files (Confirmed with HA and E2E testing Node shutdowns and recovery)

E2E HA

Project Name	CDN Update for Monomax Service	Tester		Test date	
Test Category	E2E HA	SCOPE	This test verifies the end-to-end high availability of the Monomax CDN live service by simulating node, site, and network failures and confirming uninterrupted playback, seamless failover, and proper monitoring visibility.		
Test Environment	The test environment is a multi-site CDN setup with redundant transcoders, origins, and origin shields. Live streams are ingested via SRT and UDP using HEVC with Dolby audio, with DRM enabled for HLS, DASH, and CMAF outputs. Monitoring and alerting are active across all components.		E2E Setup Test Playback From GSLB And Edge CProxy GSLB > POP > CNT > Shield > 3 x Origins Test with - Dash-Reference - Onsite Playback - Log agg. Grafana		

PART #	CATEOGRY	TEST ITEM		CONFIGURATION	HOW TO TEST	EXPECTED RESULT	TC RESULT
11	Live Origin Streamer (Transcoders+JITP+ Encryption)	EZE-HA-001	Single node failure (intra-site)	- Channel configured with 2 origin/transcoder nodes - Origin Shield configured with multi-origin failover	1) Start live channel 2) Force stop one node	- Origin Shield retries another node - Stream continues with minimal interruption - Failed node removed from active poo	- 1 Origin turned off (main-1) - Log shows traffic stops flowing to (main-1 and go to remaining 2 origin nodes) - Smooth Playback No Errors 
		EZE-HA-002	Node recovery	At least one failed node	1) Restart failed node 2) Verify node availability	- Node becomes available again to Origin Shield - Traffic gradually restored	- Stopped Origin is brought back - No stuttering or error detect (via console) 
		EZE-HA-003	Site failure (cross-site failover)	- Multi-site deployment - Origin Shield multi-origin includes both sites	1) Run channel on primary site 2) Simulate full site outage	- Origin Shield switches to secondary site nodes - Playback continues	- Main-1 and Main-2 turned off - Log shows traffic flows to only Backup-2 - No stuttering or error detect (via console) 
		EZE-HA-004	Origin restart mid-segment	HLS / DASH / CMAF output enabled	1) Restart node during segment generation	- Partial segment skipped or recovered - Player resumes at next segment	- Stopped Origin is brought back - No stuttering or error detect (via console) 
		EZE-HA-005	Origin Shield failover validation	Origin Shield configured with 2 origins per channel	1) Block origin node network 2) Observe retry behavior	- Shield retries next origin - No mass playback failure	Origin CProxy Shield content retry logic Fail ชั่ว Site Failover logic ไม่ทำ content ที่ failover CDN พอไม่เกิดปัญหา Origin ี่ 
12	Crypto	EZE-HA-006	DRM continuity on origin failover	- DRM-enabled stream - Multi-origin setup	1) Start encrypted playback 2) Trigger origin failover	- Playback continues - License remains valid	DRM streams underwent the same E2E stop-start cycle tests and no Errors were detected
		EZE-HA-007	Key rotation during failover	Periodic key rotation enabled	1) Trigger failover during key rotation	New segments use new key• Decryption succeeds	DRM streams underwent the same E2E stop-start cycle tests and no Errors were detected
13	Orign Shield	EZE-HA-008	Origin Shield node failure	Multiple shield nodes	1) Disable one shield node	Requests rerouted to healthy shield	Player showed the redirect cycle for content retries to Failover CDN failed then try other Origin Nodes. No stutter in this playback 
		EZE-HA-009	Origin Shield site failure	Shield deployed across sites	1) Simulate shield site outage	CDN fetches from alternate shield/site	All ELB and Shield in Center-1 are turned off. Log shows no traffic routed to Center-1 Player shows redirects but no stuttering or streaming issues
14	Multi-Site	EZE-HA-010	Full site A blackout	Complete stack in Site A/B	1) Blackout Site A	All traffic served from Site B	All ELB and Shield in Center-1 are turned off Log shows traffic stopped going to Center-1 Streams playable with no stutter 
		EZE-HA-011	Site recovery	Site A restored	1) Recover Site A 2) Observe traffic	Gradual traffic rebalance• No stream reset	All ELB and Shield in Center-1 are restored Streams continue without stuttering. Log shows traffic coming back to Center-1 
15	Monitoring	EZE-HA-012	Alert validation	Alerts and thresholds configured	1) Trigger node/site failures	Correct alerts fired with severity	
		EZE-HA-013	Failover event visibility	Dashboards and logs enabled	1) Perform failover 2) Review dashboards	Failover events clearly visible	
16	Origin Streamer Aging Test	EZE-HA-014	72-hour continuous streaming stability	4 origin streamer nodes (2 per site) 3 channels per node 5 ABR profiles per channel - Origin Shield multi-origin	Traffic generation: 1) Start all 6 channels 2) Run k6 + curl to continuously request manifests and segments for 72 hours via Origin Shield Visual check: 3) Play selected channels using Shaka Player	- Origin streamer runs continuously for 72h - No crash or process restart - Streams continuously available	Pending
		EZE-HA-015	Origin streamer CPU stability		1) Observe CPU usage over 72h during load	- CPU stable - No abnormal spikes	Pending
		EZE-HA-016	Origin streamer memory stability		1) Observe memory usage in Grafana during 72h test	- Memory usage stable - No memory leak trend	Pending