

# JIT vs Pre-Packaged Origin — Decision Sheet

When to use just-in-time packaging vs a pre-packaged origin. Storage math, architecture rules, 2026 vendors.

## Pre-packaged origin

~33 copies on disk per title after HLS + DASH + LL-HLS + MPEG-TS + 2 audio + Smooth legacy.  
 90-min 1080p title (5 Mbps avg) ≈ 114 GB on disk.  
 1,000 titles ≈ 114 TB ≈ \$2,622 / month (S3 Standard).  
 Cost per request at origin: minimal (read + sendfile).

## JIT origin

~6 copies on disk per title — one CMAF fMP4 master per rendition.  
 Same 90-min 1080p title ≈ 22 GB on disk.  
 1,000 titles ≈ 22 TB ≈ \$515 / month (S3 Standard).  
 Cost per cache-miss segment: real CPU (parse → seek → repackage → encrypt).

## Pick the shape — by workload

Workload	Catalogue	Latency / cache profile	Choice
Small training library	< 200 titles	Storage saving < \$200/month; static-file simplicity wins.	<b>Pre-packaged static</b>
Single live event	1 channel	Sub-2 s glass-to-glass; JIT buffer costs latency budget.	<b>Pass-through (pre-packaged)</b>
OTT VOD / SVOD	1,000+ titles	Long-tail catalogue; multiple devices and DRMs; ≥ 95% shield hit rate.	<b>JIT + origin shield</b>
Live sports (catch-up)	100+ events / month	Live + DVR + VOD; HLS + DASH + LL-HLS; manifest filtering needed.	<b>JIT + origin shield</b>
E-learning library	500+ lessons	Steady long-tail traffic; multi-language audio; protocol churn likely.	<b>JIT + origin shield</b>
Major launch / virality	1 hot title	99.99 % CDN hit rate makes storage multiplier irrelevant.	<b>Pre-packaged</b>

## 2026 JIT origin vendors

Vendor	Type	Licence model	Manifest filter	Best fit
Unified Origin	Commercial	Per-CPU-core	?filter= param	<b>Broadcasters, large OTT</b>
AWS MediaPackage v2	Managed	Ingested + egress GB	aws.manifestfilter.*	<b>AWS-stack OTT, live sports</b>
Eyevinn Smoothly + Shaka lib	OSS / DIY	Self-host	Custom	<b>Mid-size OTT, dev teams</b>
Norsk (id3as)	Commercial / OSS	Per channel	Yes	<b>Live broadcasters, e-learning</b>

## Cache-friendliness audit — run this before every JIT migration

- Origin shield enabled in front of the JIT origin (CloudFront, Akamai, Fastly, or built-in MediaPackage shield).
- Cache key includes manifest-filter query parameters (aws.manifestfilter.\*, Unified ?filter=).
- Cache key excludes session and analytics tokens (?session\_id=, ?ts=, etc.).
- Segment TTL at the edge ≥ 24 hours for VOD; ≥ segment duration for live.
- Manifest TTL ≥ several hours for VOD; equal to segment duration for live; never set to 1 s arbitrarily.
- Manifests are user-agent-agnostic — device-specific logic lives in the player, not the manifest.
- Shield hit rate observed above 90% during steady-state traffic.
- Edge hit rate observed above 95% during steady-state traffic.

## The single architectural rule for JIT

JIT origin behind a tuned origin shield behind a CDN with long segment TTLs, short manifest TTLs, manifest-filter parameters in the cache key, and session tokens excluded. Skip one of these and JIT looks twenty times more expensive than it has to be. With all of them in place, JIT typically wins by ~30x on storage vs compute.