GPU SMITH
COMPUTE ENGINEERING

GPU SMITH / RESEARCH & PRACTICAL GUIDANCE

NVIDIA NIM End of Support January 2027: Migration Plan

> Plan and build private AI compute with GPU Smith. We connect workload requirements with GPU hardware, networking, deployment and operating decisions so your infrastructure fits the work it must perform.

Published by GPU Smith · 2026-11-19 · nvidia nim end of support · nvidia nim deprecation · nim end of life migration · model-free nim · production branch 6 · nvidia ai enterprise lifecycle · private llm inference · nim vs vllm · air-gapped nim

Original article: https://gpusmith.com/articles/en/nvidia-nim-end-of-support-2027-migration-gate



In this article

1. Executive Summary
2. Introduction and Background
3. Key Changes in NVIDIA's Lifecycle Notices
4. Model-by-Model Migration Decisions
5. Implementation Considerations and Process Changes
6. Alternatives: Model-Free NIM, Other NIMs and Self-Hosted vLLM
7. Data Analysis and Evidence
8. Implications and Future Directions
9. Frequently Asked Questions (FAQs)
10. Conclusion

Executive Summary

NVIDIA's lifecycle changelog records that in September 2026 it added deprecation notices for three model-specific NVIDIA Inference Microservices (NIM) last included in Production Branch 6 (PB6), and that they reach End of Support in January 2027 (docs.nvidia.com). The three are **NIM Llama-3.1-8B-Instruct**, **NIM Llama-3.3-Nemotron-Super-49B-v1.5** and **NIM Nemotron 3 Nano**. In October 2026 the same changelog added two more PB6 artifacts, including NIM Llama-3.3-70B-Instruct (docs.nvidia.com). This report does not treat the event as a retirement of NIM as a product: the notices name specific PB artifacts, and the model-free NIM for LLMs is a migration path NVIDIA names in the notices (for Nemotron 3 Nano, the notice lists Nemotron 3.5 Lightning first).

The consequence of inaction is narrow but real. NVIDIA states that after End of Support, no further CVE patches, bug fixes, or updates will be provided for the affected PB artifact (docs.nvidia.com). Deployed containers do not stop working; older artifacts remain downloadable from the NVIDIA GPU Cloud (NGC) catalog but are no longer supported (docs.nvidia.com). PB6 itself is the last branch to include the three model-specific NIMs, and Production Branch 7 (PB7) does not (docs.nvidia.com).

Two replacement routes exist. For Llama-3.1-8B-Instruct, the documented route is the model-free NIM configured with that model source and profile (docs.nvidia.com). For Nemotron 3 Nano, NVIDIA names NVIDIA Nemotron 3.5 Lightning or the model-free NIM, and asks customers to validate application quality, performance and GPU capacity before cutover (docs.nvidia.com). The Super 49B notice adds GPU sizing to the validation list (docs.nvidia.com). A different model is therefore a possible outcome for Nano, and an open question for the 49B reasoning model that depends on the profile and model source chosen.

Three open items need written answers from NVIDIA or a reseller before a plan is final. First, the PB6 model-free NIM (version 2.0.4-pb6.0) sits in the same branch that reaches End of Life in January 2027 (docs.nvidia.com), so teams should confirm which model-free NIM version carries support afterwards. Second, production use requires an NVIDIA AI Enterprise license, with list pricing starting at $4500 per GPU per year (docs.api.nvidia.com). Third, model weights are licensed separately from the NIM container (www.nvidia.com).

The recommended posture is a dated, per-model decision: inventory image digests this week, run side-by-side quality, API and capacity tests in December, and keep a tested rollback until the vendor deadline passes. No public benchmark exists that qualifies any replacement for a given buyer's workload, so every capacity figure in this report is an arithmetic template to be replaced with measured values. As of 2026-11-19, the January 2027 deadline is under ten weeks away and the exact day is not stated by NVIDIA.

Introduction and Background

Enterprise teams that run NIM containers in private environments tend to pin an image and leave it running for months. A lifecycle notice changes that default, because support and security updates are tied to the branch an artifact belongs to, not to the container running in the cluster. This report sets out a model-by-model decision process for owners of PB6 model-specific NIM deployments: which artifacts are affected, which replacement paths NVIDIA documents, what must be tested, and how to cut over with a rollback path.

The audience is platform site reliability engineers (SREs), security teams and procurement managers. GPU Smith, an independent private AI infrastructure consultancy that advises on GPU procurement, workload sizing, inference optimization and air-gapped deployments, publishes this analysis as an adjacent advisor. It does not sell NIM, GPUs or hosting, and the NVIDIA facts in the report are cited to NVIDIA's own documentation. Package transfer for disconnected sites is covered separately in an existing air-gapped NIM upgrade runbook; this report concentrates on the support and replacement decision.

The January 2027 deadline is the reason to read this now. All NVIDIA facts below were read from NVIDIA's own pages on 2026-11-19, and the live lifecycle table should be re-checked immediately before any change is approved because NVIDIA edits it monthly; the changelog shows entries for May, July, August, September, October and November 2026.

A note on scope. Not every NIM is ending support. NVIDIA's active deprecation list is a short table of named artifacts, and other PB6 NIMs, including the model-free NIM and several model-specific NIMs, are listed in PB6 without such a notice. The analysis below stays inside the named artifacts.

Key Changes in NVIDIA's Lifecycle Notices

What the September and October 2026 notices say

The lifecycle changelog dates the first group to September 2026: three model-specific NIMs last included in PB6. The October entry adds NIM Llama-3.3-70B-Instruct and the PB-managed TAO containers, which are container flavors rather than model-specific NIMs. Each notice uses the same structure: component, current state "Deprecated", End of Support in January 2027, last supported branch PB6, first unsupported branch PB7. The Llama-3.1-8B-Instruct notice also records history: PB4 and PB5 included that model-specific NIM, and PB6 is the final branch to include it (docs.nvidia.com). The Super 49B notice records that PB5 introduced it (docs.nvidia.com), and the Nemotron 3 Nano notice records that PB6 introduced it and is the only branch to include it (docs.nvidia.com).

What "Deprecated" and "End of Support" mean

NVIDIA defines the states precisely. A Deprecated component is still supported with CVE patches and bug fixes, but a deprecation date has been set and customers have been notified (docs.nvidia.com). After End of Support, older versions stay on NGC but are unsupported, and only the latest supported version receives patches. End of Life is a stronger state: NVIDIA says customers should not use End of Life components for any workload (docs.nvidia.com). The three model-specific NIMs are in the first of these states today and move to the second in January 2027.

Reading the dates carefully matters. The notice says PB6 reaches End of Life in January 2027 and the PB artifact reaches End of Support in the same month, with no day of month given (docs.nvidia.com). Plan against the first business day of January 2027 until NVIDIA publishes a day.

Branch mechanics that explain the date

Each Production Branch is supported for nine months and receives monthly bug fixes and security updates for high and critical software vulnerabilities (docs.nvidia.com). NVIDIA releases a new Production Branch every six months (docs.nvidia.com), and NVIDIA's own cadence statement is "general guidance only" (docs.nvidia.com). The January 2027 date follows a visible pattern: the October 2025 branch (PB 25h2) reached end of life in July 2026 (docs.nvidia.com), and the May 2025 branch reached end of life in January 2026 (docs.nvidia.com). NVIDIA advises Production Branch users to apply monthly security updates and plan for major updates every six months (docs.nvidia.com). The branch guidance also lists "organizations unable to update within 9 months" as a profile for which Production Branch is not the recommended choice (docs.nvidia.com), which is relevant to teams that freeze images for long periods.

Support scope has an edge worth noting for audits. NVIDIA supports individual NIMs and software components, but not glue code (docs.nvidia.com). Application code that wraps the endpoint stays the owner's responsibility, which is why the API-behavior testing in this report is not optional.

Model-by-Model Migration Decisions

This section treats each artifact separately because the notices do. *Table 1* summarizes the affected artifacts, the vendor-named replacements and the validation work the vendor itself calls for. The "same model" column is an analytical judgment based on what the notice names, not a vendor guarantee.

Artifact	Last branch / End of Support	Vendor-named replacement	Same model preserved?	Validation owner and evidence
NIM Llama-3.1-8B-Instruct	PB6, January 2027	Model-free NIM with the Llama-3.1-8B-Instruct source and profile, or another supported model-specific NIM	Yes in principle (same weights, different packaging and runtime path)	Application owner: prompt grid and API contract; SRE: latency and VRAM; security: image digest and CVE feed

Artifact	Last branch / End of Support	Vendor-named replacement	Same model preserved?	Validation owner and evidence
NIM Llama-3.3-Nemotron-Super-49B-v1.5	PB6, January 2027	Model-free NIM with a supported model source and profile, or another supported model-specific NIM	Possible, only if a supported source and profile exist for this model in your environment	Application owner: reasoning-quality grid; SRE: GPU sizing; procurement: dual licence terms
NIM Nemotron 3 Nano	PB6 only, January 2027	Nemotron 3.5 Lightning, or model-free NIM with a supported model and profile (docs.nvidia.com)	Not if you move to Lightning (a different model)	Application owner: quality; SRE: capacity; security: new licence review
NIM Llama-3.3-70B-Instruct (October 2026 addition)	PB6, January 2027 (docs.nvidia.com)	Model-free NIM with the model source and a supported profile	Yes in principle	Adds model-source access and profile selection to the checklist (docs.nvidia.com)

The table shows that the replacement decision is not uniform. Only the 8B and 70B notices name the same model for the model-free route; the Nano notice names a different model first. The interpretive point for planners is that "same model" and "same behavior" are separate claims: a model-free container loads weights from a configured source and generates a manifest at start-up, while a model-specific container ships a curated manifest and validated profiles (docs.nvidia.com).

Llama-3.1-8B-Instruct: the most direct path

For the 8B model, NVIDIA's route is the model-free NIM configured with the Llama-3.1-8B-Instruct model source and profile. NVIDIA's own model-free documentation uses a Hugging Face path for this exact model as its example, `hf://meta-llama/Llama-3.1-8B-Instruct` (docs.nvidia.com). That repository is gated: a user must agree to share Hugging Face information to access it (huggingface.co). Credentials therefore become a deployment dependency that a model-specific container did not impose in the same way, and they must be provisioned on a connected host before an isolated site can run the model.

This is the best candidate for a same-model migration because the weights are identical and the model card documents Grouped-Query Attention (GQA) across the family (huggingface.co). Output differences can still arise from a different engine version, sampling defaults or quantization profile, so the acceptance grid below applies.

Llama-3.3-Nemotron-Super-49B-v1.5: reasoning model, heavier sizing

The Super 49B v1.5 model is described on its model card as a derivative of Meta Llama-3.3-70B-Instruct (huggingface.co) and supports a context of 128K tokens (huggingface.co). The card says its architecture work makes it possible to fit the model on a single GPU at high workloads on an H200 (huggingface.co). NVIDIA's notice says to validate model behavior, quality, performance and GPU sizing before cutover. For procurement,

the card's governing-terms section indicates NVIDIA licence terms apply to the model (huggingface.co), and because the model is derived from a Llama model, the Llama licence is relevant as well; legal review should read both texts rather than rely on this summary.

The sizing risk is sharper here than for the 8B model, because a change of engine, precision profile or tensor-parallel layout changes memory use. NVIDIA documents that raising tensor parallelism lowers the per-GPU memory floor for the models it tabulates and enables lower-memory GPUs (docs.nvidia.com), which means a profile that fits on one card may be replaced by a multi-card profile with a different cost.

Nemotron 3 Nano: a likely model change

NVIDIA's notice offers two options: Nemotron 3.5 Lightning, or the model-free NIM with a supported model and profile. Nemotron 3 Nano is a hybrid mixture-of-experts model with 3.5B active and 30B total parameters (huggingface.co). Nemotron 3.5 Lightning is catalogued with 30B total and 3B active parameters and up to 1M tokens of context (catalog.ngc.nvidia.com). These are similar in headline size but are different models, so a team that selects Lightning is changing the model, not just the container. The Lightning card states a single H100 deployment uses 256K tokens of context out of a 1M maximum (huggingface.co), and the Nano card notes a 256k default Hugging Face context because of VRAM requirements (huggingface.co). Lightning is published under a different licence, OpenMDW version 1.1 (huggingface.co), so a legal review of the new model is a prerequisite and not an afterthought.

Related artifacts that share the deadline

The same deadline applies to the Llama-3.3-70B-Instruct model-specific NIM and the E5 embedding NIM (for which NVIDIA names NVIDIA Retrieval QA Llama 3.2 1B Embedding v2 as the transition target), which stops receiving CVE patches after deprecation (docs.nvidia.com). Separately, NVIDIA records that the TensorRT-LLM backend container variant is not included in PB6, with a transition path to the vLLM backend Triton variant (docs.nvidia.com); and an older 70B NIM (Llama-3.1-70b-instruct) had its updates end after July 2026 (docs.nvidia.com). Fleet owners should scan for these as well, because they surface in the same inventory.

Implementation Considerations and Process Changes

Step 1: Inventory what is deployed

Start with facts, not assumptions. For every running NIM, record the registry reference, the image digest, the branch indicator in the tag, the model name and the profile in use. Kubernetes documents that tags can be moved to point to different images, but digests are fixed (kubernetes.io). If a reference carries both tag and digest, only the digest is used for pulling (kubernetes.io). Pinning by digest prevents a registry-side change from producing a mix of versions across Pods (kubernetes.io). The inventory sheet should therefore be keyed on digest, with the tag as a human label.

NVIDIA marks Production Branch NIM names with a branch suffix, as in the `llm-nim-pb25h2` container, which NGC describes as releasing every six months with a three-month overlap between releases (catalog.ngc.nvidia.com). That overlap is a practical planning cue: a new branch overlaps the old one for a limited window, so the replacement must be qualified inside the overlap, not after it.

The inventory also feeds supply-chain records. The National Institute of Standards and Technology (NIST) Secure Software Development Framework describes a core set of high-level secure software development practices (csrc.nist.gov), and its AI community profile adds practices specific to AI model development (csrc.nist.gov). The Cybersecurity and Infrastructure Security Agency (CISA) frames a software bill of materials as a detailed inventory of software components (www.cisa.gov). NIST's container guide covers the security concerns associated with using containers (csrc.nist.gov). These documents do not mandate a NIM migration, but they supply vocabulary for the inventory a security team will want on file.

Step 2: Check entitlement and licensing

Production use of NIM requires an NVIDIA AI Enterprise license (docs.api.nvidia.com). NVIDIA states that the software is licensed on a per-GPU basis (docs.nvidia.com), that pricing is by GPU count and not by number of NIMs (docs.api.nvidia.com), and that list prices start at $4500 per GPU per year, or about $1 per GPU per hour in the cloud. Because the price is per GPU, replacing three model-specific containers with model-free containers on the same GPUs does not by itself change the licence count; it can change it if the new profile needs more GPUs.

Support terms matter at the margin: subscription support is included with the annual licence cost (docs.nvidia.com), and the Business Standard tier lists a 4-hour initial response, with coverage during local business hours (8 am-5 pm, excluding observed holidays) (docs.nvidia.com). Some H100 PCIe and NVL boards include a five-year subscription (docs.nvidia.com), so an older purchase may already carry entitlement; check serial numbers and activation status. Developer Program access is described for research, development and experimentation on up to 16 GPUs (docs.api.nvidia.com), and it does not include enterprise features or support (docs.api.nvidia.com). A production cluster running on developer access has no support entitlement to preserve.

Two scope limits affect audit evidence. NVIDIA's support for NIM covers the runtime and not model outputs or the models themselves (docs.api.nvidia.com), and the models are licensed separately from the NIM container. Llama weights carry Meta's terms, including a separate-licence requirement above 700 million monthly active users (www.llama.com) and an obligation to adhere to the Acceptable Use Policy (www.llama.com).

Step 3: Understand API, profile and cache behavior

NIM exposes OpenAI-compatible endpoints; the chat completions endpoint supports streaming and tool calling (docs.nvidia.com). The documented NIM LLM and VLM line now packages open-source engines, and each container image includes one engine (docs.nvidia.com). Release 2.1.6 upgraded the bundled backends to vLLM 0.32.0 and SGLang 0.5.22 (docs.nvidia.com). NVIDIA's earlier multi-LLM container selected among TensorRT-LLM, vLLM and SGLang (huggingface.co), so teams coming from older images may see a backend change that alters latency distributions and structured-output behavior even when the model is the same.

Profile selection is a second change surface. At container startup NIM selects exactly one profile from the manifest (docs.nvidia.com), following a priority chain evaluated top to bottom where the first match wins (docs.nvidia.com). NIM estimates the GPU memory each profile requires and classifies profiles into compatible categories (docs.nvidia.com). For model-free NIM the manifest is generated at runtime, which is why pinning

an explicit profile (rather than relying on automatic choice) belongs in the runbook. Some models have profiles built for specific GPU model and count, with no generic fallback (docs.nvidia.com), so a hardware change can remove the supported profile.

The cache is the third surface. NIM stores model and artifact files in a cache directory inside the container (docs.nvidia.com). With the Kubernetes NIM Operator, caching avoids each instance downloading its own copy (docs.nvidia.com), and NVIDIA recommends profile filtering when caching so the cache job does not fetch more than intended (docs.nvidia.com). Re-warm and re-validate the cache after changing images; do not assume the old model store is reusable.

Step 4: Side-by-side quality, API, latency and VRAM tests

NVIDIA's benchmarking guide puts accuracy evaluation outside its scope, to be validated separately for your use case (docs.nvidia.com). This means quality evidence has to come from the owner's own prompts. *Table 2* sets out an acceptance grid that can be filled in per model. The thresholds are placeholders: they must come from the application's service-level objectives (SLOs) and risk owners, and this report does not assert any value as a standard.

Dimension	How to test	Evidence to retain	Example acceptance rule (placeholder)
Application quality	Replay a frozen prompt set (production-sampled, de-identified) through old and new endpoints; score with a fixed rubric or reference answers	Prompt set hash, scorer version, per-category scores	New score is not below old score minus a tolerance set by the application owner
API behavior	Contract tests on the request fields your clients send: streaming, tool calls, stop tokens, `max_tokens`, error codes	Test log with request and response fixtures	Zero contract differences, or each difference accepted in writing
Latency	Concurrency sweep with time to first token (TTFT) and inter-token latency (ITL), same tool and definitions on both sides (docs.nvidia.com)	Tool version, command line, raw output	p95 TTFT and ITL within the SLO at target concurrency
Throughput and VRAM	Find the concurrency where throughput saturates (docs.nvidia.com); record peak memory	Latency-throughput curve, peak GPU memory	Headroom ratio below the agreed margin (see the data section)
Security and support	Record image digest, signature check, scan results and CVE feed owner	Digest list, scan reports	Every running image maps to a supported artifact
Rollback	Redeploy the previous digest in staging and run the smoke suite	Rollout history and timing	Previous version restored within the agreed window

The grid separates concerns that are often blurred. A latency pass says nothing about answer quality, and a quality pass says nothing about whether the cluster survives peak load. NVIDIA warns that tools differ on whether TTFT is included in the ITL average (docs.nvidia.com), and that time to first token generally includes queuing, prefill and network time (docs.nvidia.com). Use the same benchmark client and version on both sides, and keep in mind that single-process load generators can introduce client-side queuing bottlenecks at high concurrency (arxiv.org).

Prefer concurrency-based load to fixed request rates: with a fixed rate above capacity, outstanding requests grow without bound (docs.nvidia.com). Compare model-specific and model-free runs with prefix caching settings made explicit, because vLLM's automatic prefix caching reuses the KV cache of earlier queries that share a prefix (docs.vllm.ai), which can flatter results on repetitive prompts.

Step 5: Connected versus air-gapped preparation

In a connected environment, a remote model path needs network access and credentials on the first deployment (docs.nvidia.com), then the generated manifest is reused afterwards without outbound network calls (docs.nvidia.com). In an air-gapped environment, the simplest documented workflow is a local directory: NIM generates the manifest from the local directory with no network access or credentials (docs.nvidia.com). For model-specific and remote sources, NVIDIA's air-gap guidance has a connected staging phase in which the download-to-cache command stores selected or default model profiles in the NIM cache (docs.nvidia.com). For Kubernetes, the NIM Operator documents running with no connection to the NGC registry or Hugging Face Hub through proxy or mirrored registries (docs.nvidia.com). Every image referenced by a Helm chart (init containers and sidecars included) must also be mirrored; the 2.0.13 air-gap page gives the example of an init container image that defaults to a public base image. Gated weights (such as Llama) must be downloaded by an authorized account on the connected side and transferred as an approved artifact, with the transfer logged. The existing GPU Smith runbook referenced above covers the transfer mechanics.

Step 6: Staged cutover, rollback and audit evidence

Kubernetes Deployments make staged cutover straightforward. The default rolling update keeps at least 75% of desired Pods up (kubernetes.io) and allows up to 125% of desired Pods during the update (kubernetes.io); for GPU-bound inference, the surge Pods need free GPUs, so the default surge may be impossible on a full cluster and the strategy should be set explicitly. Rollback depends on retained history: old ReplicaSets beyond the history limit are garbage-collected (kubernetes.io), and a chosen revision can be restored with `kubectl rollout undo` and `--to-revision` (kubernetes.io). A rollout that does not progress is reported failed after a default 600 seconds (kubernetes.io), which may be too short for a large model load, so set the deadline from measured cold-start time. If the NIM Operator manages the service, a NIMService creates a Kubernetes Deployment, so the same mechanics apply (docs.nvidia.com).

Audit evidence should be a short pack: the inventory with digests, the filled acceptance grid, the licence and entitlement check, the approvals, the rollout record and the rollback test result. NIST describes its AI Risk Management Framework as released on 26 January 2023 and voluntary (www.nist.gov), which makes it a reasonable vocabulary source for the risk sign-off, not a requirement.

Table 3 sets out a planning calendar from this report's publication date. The dates are planning assumptions, not vendor dates; the only vendor date is "January 2027".

Window (2026 to 2027)	Action	Exit criterion
Nov 19 to Nov 27	Inventory digests, tags, profiles, models; confirm entitlement and licence texts	Inventory signed off; every PB6 model-specific NIM mapped to a destination

Window (2026 to 2027)	Action	Exit criterion
Nov 30 to Dec 11	Build test rig; run quality, API and concurrency tests on old and new paths	Acceptance grid filled for each model
Dec 14 to Dec 18	Decide per model: model-free same model, different model, or hold with documented risk acceptance	Decision memo per artifact
Dec 21 to Jan 8	Canary 5 to 10% traffic where traffic changes are possible; stage air-gapped artifacts in parallel	No SLO regression for a full business cycle
Jan 11 to Jan 22	Full cutover; keep prior digest and cache for rollback	Rollback tested within the window
After NVIDIA's stated day	Retire old image from registries or tag it unsupported; archive evidence	Zero running unsupported PB artifacts, or a signed exception

The calendar compresses work into the holiday period, so teams with change freezes should move the decision memo earlier. The explicit exception row exists because some regulated environments cannot change within the window, and an exception should be signed by the risk owner, with compensating controls such as network isolation and scanning.

Alternatives: Model-Free NIM, Other NIMs and Self-Hosted vLLM

The replacement options form a spectrum. The nearest is a model-free NIM from NVIDIA, which lets a team run any supported model without a model-specific container image (docs.nvidia.com). Its limits are explicit: if the architecture is unsupported by the container's inference backend, NIM cannot serve it regardless of where the model is hosted (docs.nvidia.com). The vLLM model-free container can serve any model the bundled vLLM version supports beyond the validated list (docs.nvidia.com), and NVIDIA lists the GPUs on which it has been verified (docs.nvidia.com); "can serve" is a capability statement and not a support statement for a given model.

The second option is a different supported model-specific NIM, which the notices list as the alternative route. The third is running an open-source engine directly. vLLM documents more than 200 supported model architectures on Hugging Face (docs.vllm.ai) and an OpenAI-compatible server with additional Anthropic Messages API and gRPC support (docs.vllm.ai). NVIDIA's TensorRT-LLM also ships an OpenAI-compatible server command (nvidia.github.io), and NVIDIA Dynamo is documented as coordinating pipelines with features such as disaggregated serving, KV-aware routing and autoscaling (docs.nvidia.com).

A direct vLLM deployment moves responsibilities to the operator. The vLLM security guide says not to rely exclusively on the API key to secure access (docs.vllm.ai) and states that communication between nodes in a multi-node deployment is insecure by default (docs.vllm.ai). Advisories are published through GitHub using the same system used for reporting (docs.vllm.ai). A self-hosted stack has no NVIDIA support entitlement attached, so the team must own patch tracking. The trade is flexibility and portability against an assigned support contact.

No independent head-to-head benchmark can decide this for a buyer. Vendor and author results exist, such as the reported 2-4x throughput improvement at the same latency in the PagedAttention paper's own comparison (arxiv.org) and up to 6.4x in SGLang's paper against the systems it chose to compare (arxiv.org), but these

were measured on specific models, hardware and years. A 2026 benchmarking framework paper concludes there is no universal best option because tradeoffs depend on workload and constraints (arxiv.org). MLCommons restricts its "Available" category to components that can be bought or rented (mlcommons.org), which is a useful filter for procurement, but published MLPerf results do not substitute for testing a particular prompt mix.

Data Analysis and Evidence

This section turns the cited facts into a sizing template. Every derived number is arithmetic on a cited input, labeled as such, and none is a measurement.

What the lifecycle data show

NVIDIA's active deprecation table lists five NIM entries with January 2027 dates: four model-specific LLM NIMs (8B, 49B, Nano and 70B) and the E5 embedding NIM. Three of the four LLM NIMs were added in September and the 70B in October (alongside the TAO containers), so the notice cadence is about one batch a month. The branch pattern gives a second data point: a nine-month support window with a new branch every six months implies a three-month overlap, which the NGC page for the PB container describes directly. The previous branch ended in July 2026 (catalog.ngc.nvidia.com), so the observed gap between branch end dates is six months.

Memory arithmetic for the 8B model

Memory sizing starts with weights. NVIDIA's NIM 1.8.0 guide gave about 15 GB for a Llama 8B model (docs.nvidia.com), with the caveat that the numbers are a rough guideline (docs.nvidia.com). That is consistent with 8 billion parameters at 2 bytes each (16 GB in decimal units). Weights are only part of the budget. The PagedAttention paper's KV cache formula multiplies two vectors, hidden size, layers and bytes per value, and its OPT-13B example shows that a 2048-token request can use as much as 1.6 GB (arxiv.org). It also reports that, for a 13B model on a 40 GB A100, close to 30% of memory stores dynamic request state (arxiv.org).

For Llama 3 models, the architecture paper lists 8 key-value heads for all sizes (arxiv.org), using grouped-query attention to shrink KV caches during decoding (arxiv.org). Using the paper's 8B figures of 32 layers and a head dimension of 128 (derived as 4096 divided by 32 heads), per-token KV cache in 16-bit precision is 2 (K and V) x 32 layers x 8 heads x 128 x 2 bytes = 131,072 bytes, or 128 KiB. A single 8,192-token sequence therefore holds 1 GiB of KV cache, and a single 131,072-token sequence holds 16 GiB. These are derived values; the layer count and head dimension should be confirmed against the model's configuration file.

NVIDIA's NIM support matrix makes the same point for newer models: minimum memory figures cover weights and runtime allocations, and do not include headroom for large KV caches at long context or high concurrency (docs.nvidia.com).

GPU memory reference points

Capacity references from NVIDIA's own product pages: the L40S lists 48GB GDDR6 with ECC (www.nvidia.com), the RTX PRO 6000 Blackwell Server Edition lists 96GB GDDR7 with ECC (www.nvidia.com), and the H200 lists 141GB (www.nvidia.com). The Nano card names H100-80GB and A100

as supported hardware (huggingface.co), and the FP8 variant of Nano quantizes the KV cache as well (huggingface.co), which changes the memory profile and requires its own quality test. Where vLLM sizes memory by estimate, the utilization default is 0.92 (docs.vllm.ai); confirm the value for the version the container bundles.

Worked example: headroom calculation (Hypothetical Example)

The buyer-supplied inputs below are invented to show the method. An internal assistant on an 8B model has a measured peak of 40 concurrent requests and an observed saturated capacity of 55 concurrent requests at the latency SLO on the candidate profile.

- **Headroom ratio** = peak measured concurrency demand divided by observed capacity = 40 / 55 = 0.73.
- **Margin policy** (buyer-defined): keep the ratio at or below 0.70 so failover and growth fit. 0.73 fails, so the options are another GPU, a lower context limit, or quantization (which requires re-running the quality grid).
- **KV check**: at an average of 8,192 tokens per active sequence, 40 sequences hold about 40 GiB of KV cache using the 128 KiB per token figure above. On an 80 GB card (about 74.5 GiB) with about 15 GiB of weights (roughly 16 GB), that is about 55 GiB (roughly 59 GB) before runtime overhead, which leaves little headroom and fits only if the engine's memory-utilization setting leaves room.

The concurrency link is Little's law, which states that mean number in system equals arrival rate times mean time in system when the means are finite and the processes stationary (econpapers.repec.org). A university lecture note adds the assumption that system capacity is sufficient for the arrivals (www.win.tue.nl). Both assumptions fail at saturation, which is why capacity must be measured on the buyer's own hardware and prompt lengths instead of computed from vendor numbers.

Implications and Future Directions

For fleet owners the lasting change is that a model-specific NIM is a lifecycle-bound artifact. NVIDIA's own wording is that models are packaged in containers that ship a model-specific manifest and curated weights, while model-free containers serve a model configured at runtime. The first is easier to qualify as a unit; the second turns the model source, credentials and profile into configuration that the operator must govern. Expect the model-free approach to become the standing pattern, and plan the configuration management accordingly.

For security teams, the vendor boundary is clear: patch and bug-fix obligations stop at End of Support for the named artifact. NVIDIA publishes security bulletins through its product security pages and advises customers to subscribe to notifications (www.nvidia.com); it states that bulletins are published in Markdown, CSAF and CVE formats (www.nvidia.com), and rates severity with CVSS version 3.1 (www.nvidia.com). Map each running digest to a feed that will cover it after January.

For procurement, the replacement may change licence surfaces more than the NVIDIA subscription. If Nano becomes Lightning, a new model licence enters the file. If a hardened deployment was chosen because most PB6 software is available as FIPS-enabled Government Ready versions (docs.nvidia.com), confirm that the replacement is available in that form too, because the notice text does not address it.

Open questions remain. NVIDIA's pages read for this report do not give a day for End of Support, do not state a PB7 release date, and do not state which model-free NIM release carries support beyond PB6. The latest NIM LLM docs show a 2.x line with its own cadence, so teams should ask NVIDIA or their reseller which release they should standardize on. Also, documentation changes: the fetched NIM docs describe environment variables and profiles for the 2.x line, and 1.x selectors such as automatic backend priority are not carried forward, so old runbooks need review.

Frequently Asked Questions (FAQs)

Is NVIDIA ending support for all NIMs in January 2027?

No. The notices name specific artifacts: three model-specific NIMs added in September 2026, plus two PB6 artifacts added in October. Other PB6 NIMs appear in the PB6 listing without such a notice.

Which NIM microservices are being deprecated?

Llama-3.1-8B-Instruct, Llama-3.3-Nemotron-Super-49B-v1.5 and Nemotron 3 Nano, then Llama-3.3-70B-Instruct and the E5 embedding NIM, all with January 2027 dates.

Will deployed containers stop working?

Nothing in the notice says containers stop running. Older artifacts remain on NGC but are unsupported. The loss is patches, fixes and updates.

How to migrate off a deprecated self-hosted NIM?

Inventory digests, choose per model between the model-free NIM and a different supported model, test quality, API, latency and VRAM side by side, stage artifacts for disconnected sites, cut over with a rollback, and keep evidence. The steps above give the sequence.

NIM versus vLLM for self-hosted inference?

NIM packages open-source engines with NVIDIA support under an AI Enterprise licence; running vLLM directly puts patch tracking and hardening on the operator. Neither choice removes the need for your own tests.

Does this change the NVIDIA AI Enterprise support lifecycle generally?

The policy is unchanged: nine-month Production Branches with monthly security updates and a new branch every six months.

Conclusion

The January 2027 notices are a bounded, dated event with a known shape: three model-specific NIMs from PB6, plus later PB6 additions, lose support when the branch ends. The replacement is not a single answer. The 8B model has a documented same-model route through the model-free NIM, the 49B reasoning model

depends on a supported source and profile and needs a fresh sizing exercise, and Nano may become a different model. The right deliverable for each artifact is a decision memo backed by a filled acceptance grid, a digest inventory, an entitlement check and a tested rollback.

Because compatibility cannot be promised from documentation, the evidence must come from the buyer's own prompts, traffic and GPUs. Re-read NVIDIA's lifecycle table before sign-off, confirm the model-free NIM release that will carry support after PB6, and treat every figure in this report that is derived arithmetic as a template to be replaced by measurement.

Source links

Links cited in this article, in order of first appearance. Full addresses are provided for readers using extracted text.

1. https://docs.nvidia.com/ai-enterprise/lifecycle/latest/index.html
2. https://docs.nvidia.com/ai-enterprise/lifecycle/latest/eol-notices.html
3. https://docs.nvidia.com/ai-enterprise/lifecycle/latest/application-software.html
4. https://docs.api.nvidia.com/nim/docs/product
5. https://www.nvidia.com/en-us/agreements/enterprise-software/nvidia-software-licensing-overview-and-faq
6. https://docs.nvidia.com/ai-enterprise/lifecycle/latest/eol-branches.html
7. https://docs.nvidia.com/ai-enterprise/lifecycle/latest/choosing-a-branch.html
8. https://docs.nvidia.com/ai-enterprise/lifecycle/latest/lifecycle-policy.html
9. https://docs.nvidia.com/nim/large-language-models/latest
10. https://docs.nvidia.com/nim/large-language-models/latest/deployment/model-free-nim.html
11. https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct
12. https://huggingface.co/nvidia/Llama-3_3-Nemotron-Super-49B-v1_5
13. https://docs.nvidia.com/nim/large-language-models/2.0.13/reference/support-matrix.html
14. https://huggingface.co/nvidia/NVIDIA-Nemotron-3-Nano-30B-A3B-BF16
15. https://catalog.ngc.nvidia.com/orgs/nim/nvidia/containers/nemotron-3.5-lightning-30b-a3b
16. https://huggingface.co/nvidia/NVIDIA-Nemotron-3.5-Lightning-30B-A3B-BF16
17. https://kubernetes.io/docs/concepts/containers/images
18. https://catalog.ngc.nvidia.com/orgs/nvidia/-/containers/llm-nim-pb25h2/-
19. https://csrc.nist.gov/pubs/sp/800/218/final
20. https://csrc.nist.gov/pubs/sp/800/218/a/final
21. https://www.cisa.gov/resources-tools/resources/2025-minimum-elements-software-bill-materials-sbom
22. https://csrc.nist.gov/pubs/sp/800/190/final
23. https://docs.nvidia.com/ai-enterprise/planning-resource/licensing-guide/latest/licensing.html
24. https://www.llama.com/llama3_1/license
25. https://docs.nvidia.com/nim/large-language-models/latest/reference/api-reference.html
26. https://docs.nvidia.com/nim/large-language-models/latest/introduction.html
27. https://docs.nvidia.com/nim/large-language-models/latest/release-notes.html

28. https://huggingface.co/blog/nvidia/multi-llm-nim
29. https://docs.nvidia.com/nim/large-language-models/latest/deployment/model-profiles-and-selection.html
30. https://docs.nvidia.com/nim/large-language-models/latest/reference/environment-variables.html
31. https://docs.nvidia.com/nim-operator/latest
32. https://docs.nvidia.com/nim-operator/latest/cache-llm.html
33. https://docs.nvidia.com/nim/benchmarking/llm/latest/index.html
34. https://docs.nvidia.com/nim/benchmarking/llm/latest/metrics.html
35. https://docs.nvidia.com/nim/benchmarking/llm/latest/parameters.html
36. https://arxiv.org/abs/2605.24217
37. https://docs.vllm.ai/en/latest/features/automatic_prefix_caching.html
38. https://docs.nvidia.com/nim/large-language-models/latest/deployment/air-gap-deployment.html
39. https://docs.nvidia.com/nim-operator/3.1.2/air-gap.html
40. https://kubernetes.io/docs/concepts/workloads/controllers/deployment
41. https://www.nist.gov/itl/ai-risk-management-framework
42. https://docs.nvidia.com/nim/large-language-models/latest/support-matrix.html
43. https://docs.vllm.ai/en/latest
44. https://nvidia.github.io/TensorRT-LLM/commands/trtllm-serve/trtllm-serve.html
45. https://docs.nvidia.com/dynamo/latest
46. https://docs.vllm.ai/en/latest/usage/security.html
47. https://docs.vllm.ai/en/latest/contributing/vulnerability_management.html
48. https://arxiv.org/abs/2309.06180
49. https://arxiv.org/abs/2312.07104
50. https://arxiv.org/abs/2511.16682
51. https://mlcommons.org/benchmarks/inference-datacenter
52. https://catalog.ngc.nvidia.com/orgs/nvidia/-/collections/production_branch_october_2025/-
53. https://docs.nvidia.com/nim/large-language-models/1.8.0/getting-started.html
54. https://arxiv.org/pdf/2309.06180
55. https://arxiv.org/pdf/2407.21783
56. https://www.nvidia.com/en-us/data-center/l40s
57. https://www.nvidia.com/en-us/data-center/rtx-pro-6000-blackwell-server-edition
58. https://www.nvidia.com/en-us/data-center/h200
59. https://huggingface.co/nvidia/NVIDIA-Nemotron-3-Nano-30B-A3B-FP8
60. https://docs.vllm.ai/en/latest/configuration/optimization
61. https://econpapers.repec.org/RePEc:inm:oropre:v:9:y:1961:i:3:p:383-387
62. https://www.win.tue.nl/~iadan/2V050/h2.pdf
63. https://www.nvidia.com/en-us/security
64. https://www.nvidia.com/en-us/security/psirt-policies

THE PUBLISHER

About GPU Smith

GPU Smith provides independent engineering and research for private AI infrastructure. We help technical buyers and operators reason about GPU workload sizing, hardware selection, procurement, deployment and inference operations, from the compute system to the networking, power and cooling requirements around it.

Start with the workload

Useful infrastructure decisions begin with the models, data, throughput, latency and operating constraints a team actually has. GPU Smith helps connect those requirements with system design and deployment choices. The goal is a privately controlled AI environment whose capacity and operational demands are understood before a purchasing decision.

Hardware and supplier research

Our public reference library covers GPUs, complete systems and networking components. The server-vendor directory and manufacturing research help teams investigate suppliers, compare options and follow sources behind technical claims. We distinguish manufacturer specifications from measured benchmarks and advertised capabilities from tested configurations. Reference pages are research resources, not a live inventory listing or a binding equipment quote.

Deployment and operations

GPU Smith's engineering scope includes infrastructure integration, networking, power, cooling and the practical operation of inference workloads. Our published articles explain the tradeoffs behind deployment, procurement and ongoing operations so teams can ask better questions and document decisions.

Work with GPU Smith

Explore the hardware library, GPU references, networking references and vendor research. Contact GPU Smith with your workloads and deployment constraints to discuss a project and confirm current scope, pricing and availability.

Inclusion of a manufacturer or product in our research does not imply a partnership, certification or endorsement.

Website: https://gpusmith.com

This article is educational. Verify consequential medical, legal, financial, regulatory and technical decisions with appropriate professionals and the cited primary sources. Company services are described separately from the article's research findings.