GPU SMITH
COMPUTE ENGINEERING

GPU SMITH / RESEARCH & PRACTICAL GUIDANCE

AWS G6f Fractional vs Whole L4 Break-Even Analysis

> 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 · 2026-09-19 · aws g6f fractional l4 vs whole l4 · aws g6f pricing · fractional gpu · nvidia l4 · aws g6 · gpu inference · finops · model serving · gpu cost

Original article: https://gpusmith.com/articles/en/aws-g6f-fractional-vs-whole-l4

In this article

Executive Summary

For a portfolio of twelve low-volume inference endpoints, **AWS G6f fractional L4 instances are not automatically cheaper than whole-L4 G6 instances**. As of September 19, 2026, Linux On-Demand rates in US East (N. Virginia) were **$0.202/hour** for g6f.large, **$0.475/hour** for g6f.2xlarge, **$0.950/hour** for g6f.4xlarge, and **$0.8048/hour** for the full-L4 g6.xlarge b0.p.awsstatic.com b0.p.awsstatic.com. The comparison is not one eighth of a GPU price versus one whole GPU price. Each shape bundles a different quantity of central processing unit (CPU), system memory, local storage, network capacity, and operational isolation.

Using the question's portfolio, four nominal 3 GB endpoints, four 6 GB endpoints, and four 12 GB endpoints, a **single-replica G6f floor costs $6.508/hour**, or **$4,750.84 per 730-hour month**, if g6f.large supplies enough host resources. Four whole g6.xlarge hosts cost **$3.2192/hour**, or **$2,350.02 per month**, and have enough aggregate guest-visible memory for one carefully measured mixed placement. These are capacity-model outputs, not benchmark results. AWS reports 2.79 GiB, 5.59 GiB, 11.18 GiB, and 22 GiB of accelerator memory for the relevant guest profiles, below the marketed 3 GB, 6 GB, 12 GB, and 24 GB labels docs.aws.amazon.com aws.amazon.com. A model that merely fits its artifact label can still fail at warmup or maximum context because TensorRT algorithms may require GPU workspace and key-value (KV) cache consumes separate per-GPU memory docs.nvidia.com docs.vllm.ai

The decision rule is therefore: **keep an endpoint on CPU if CPU meets its quality, availability, and P95 latency service-level objective (SLO); choose the smallest validated G6f profile when isolation and deployment independence dominate; consolidate on whole G6 only after a mixed-trace test proves memory, latency, fairness, failure, and rollout behavior**. Dynamic batching often improves throughput, but NVIDIA says its benefit is model-specific docs.nvidia.com. The whole-GPU case remains cheaper until the consolidated design needs **nine g6.xlarge hosts** for each one-replica portfolio, since eight cost $6.4384/hour and nine cost $7.2432/hour. High availability, correlated peaks, isolation rules, and engineering time can move that threshold. The correct output is a placement policy with remeasurement triggers, not a universal GPU winner.

Introduction and Background

The search for the **cheapest AWS GPU for inference** often starts with the fraction printed beside an accelerator. That shortcut fails for small AI endpoints because cloud billing occurs at instance granularity, while service reliability occurs at endpoint and replica granularity. AWS launched G6f as fractional NVIDIA L4 capacity with one-eighth, one-quarter, and one-half profiles, and allows On-Demand, Spot, or Savings Plans purchasing aws.amazon.com The full **NVIDIA L4** is a 24 GB GDDR6, single-slot physical accelerator www.nvidia.com.

The physical card's product brief calls for Linux driver R525 or later www.nvidia.com. That full-card requirement is only a starting point because AWS documents a narrower GRID branch for the fractional environment.

This report treats the proposed twelve endpoints as a capacity-planning portfolio. Each endpoint must have its own artifact hash, precision, request trace, quality gate, P95 latency SLO, availability target, startup time, and isolation class. Average latency is insufficient because it can hide tail latency sre.google The comparison measures **SLO-compliant endpoint-hours**, not accelerator-hours alone.

In practical search terms, this is an **AWS G6f instance pricing** and **AWS G6f fractional GPU pricing** analysis, plus an AWS G6f break-even analysis. It directly answers “fractional L4 GPU vs full L4 GPU,” “NVIDIA L4 cost per hour AWS,” and “AWS GPU instances for small AI endpoints” as implementation questions. It also identifies the cheapest AWS GPU for inference and when fractional GPU inference cost beats a shared full GPU.

GPU Smith is adjacent to this purchasing decision, not an AWS or NVIDIA product vendor. Its stated practice includes “Capacity planning, telemetry, failure-mode analysis and operating procedures” gpusmith.com That perspective supports a neutral rule used throughout this analysis: assumptions become placement decisions only after acceptance tests. No public benchmark can determine how these twelve private request traces interact.

Architecture A: One Fractional G6f VM per Endpoint

Capabilities

AWS maps nominal profiles to concrete instances: g6f.large and g6f.xlarge expose one-eighth and 3 GB, g6f.2xlarge exposes one-quarter and 6 GB, and g6f.4xlarge exposes one-half and 12 GB docs.aws.amazon.com docs.aws.amazon.com. The one-eighth choices differ in host resources, so g6f.large is not interchangeable with g6f.xlarge merely because the accelerator slice matches.

AWS's detailed table reports **2.79 GiB** visible accelerator memory for g6f.large and g6f.xlarge, **5.59 GiB** for g6f.2xlarge, and **11.18 GiB** for g6f.4xlarge docs.aws.amazon.com. That is the safer initial ceiling for guest-side admission, followed by an actual `nvidia-smi` check on the selected image.

The endpoint inventory should record:

- **Identity:** model name, immutable artifact hash, framework, serving-engine version, and container digest.

- **Numerics:** precision, quantization method, and any quality delta against the approved baseline.
- **Memory:** resident weights, post-warmup allocation, maximum allocation, KV cache, framework workspace, and buyer-required reserve.
- **Host demand:** measured CPU, random-access memory (RAM), ephemeral storage, Elastic Block Store (EBS), and network throughput.
- **Traffic:** timestamped request trace, payload and context distribution, peak concurrency, and cross-endpoint correlation.
- **SLOs:** quality threshold, P50 and P95 latency, error ceiling, availability target, and recovery time.
- **Operations:** startup and model-load time, replica count, Availability Zone (AZ) policy, and isolation class.

PyTorch exposes current and maximum tensor-memory counters, which makes runtime allocation measurable independently from artifact size docs.pytorch.org. The profile passes only when `visible memory minus measured peak minus reserve` is nonnegative during the maximum tested context, batch, and concurrency.

That reserve must include more than tensor allocations. vLLM reports KV cache as a distinct per-GPU byte allocation docs.vllm.ai, while continuous batching trades larger input buffers against fewer KV-cache blocks huggingface.co. A profile that passes with short prompts can therefore fail the approved maximum context.

Adoption

AWS lists G6f in US East (N. Virginia), but region listing is not proof of capacity in a selected AZ docs.aws.amazon.com. The default account quota for running On-Demand G and VT instances is documented as zero vCPUs and adjustable, so quota readiness belongs in the launch plan docs.aws.amazon.com.

Amazon Elastic Container Service (ECS) can place fractional and full GPU instances in the same capacity provider docs.aws.amazon.com. This enables a mixed portfolio, but does not eliminate per-instance rounding.

Strengths and Limitations

Strengths are simple attribution, per-endpoint scaling, small failure domains, and direct isolation boundaries. A bad deployment affects one endpoint replica instead of several colocated models. Per-endpoint telemetry and rollback are also easier to reason about.

Limitations include twelve minimum billable instances for one replica each, or twenty-four for two replicas each. Auto Scaling maintains a configured minimum and replaces impaired instances to preserve desired capacity docs.aws.amazon.com That reliability feature preserves idle cost as well as availability.

G6f also has a specific software surface. AWS requires **GRID 18.4 through 19.5** for G6f and Gr6f, while full G6 accepts a different documented range docs.aws.amazon.com NVIDIA warns that CUDA feature support can vary by virtual-GPU software version docs.nvidia.com Every framework, custom kernel, container, and operator therefore needs validation on G6f, even if it works on the same underlying L4 silicon in G6.

Container compatibility also has a hard lower bound: NVIDIA container images can reject a driver that is insufficient for their CUDA version docs.nvidia.com. Pinning and testing the complete image is safer than testing a framework name alone.

Architecture B: Consolidated Whole-L4 G6

Capabilities

The smallest full-GPU option in this comparison, **g6.xlarge**, exposes 4 vCPUs, 16 GiB system memory, one L4, and **22 GiB** of accelerator memory in AWS's detailed specification docs.aws.amazon.com. Its catalog record includes one 250 GB NVMe solid-state drive and networking up to 10 Gigabit b0.p.awsstatic.com.

Consolidation requires a serving control plane, not only a memory spreadsheet. NVIDIA Triton supports TensorRT, PyTorch, Open Neural Network Exchange (ONNX), OpenVINO, Python, and other backends docs.nvidia.com Its explicit model-control mode makes all post-start load and unload actions deliberate docs.nvidia.com.

Triton's repository is file-system based docs.nvidia.com. Artifact promotion, warmup, and rollback must therefore cover repository state as well as a deployment manifest.

A production whole-L4 layer needs:

- **Memory admission:** reject a placement whose measured peak plus reserve exceeds the host budget.
- **Load policy:** define eager versus explicit load, eviction order, warmup, and recovery behavior.
- **Concurrency control:** cap each model's inflight work and batch delay.
- **Fairness:** prevent a busy endpoint from consuming the latency budget of a quiet one.
- **Routing:** keep replica and AZ placement visible to the request router.
- **Observability:** record queue time, execution time, GPU memory, utilization, errors, and per-model P95 latency.
- **Change control:** use canaries, surge capacity, health checks, and deterministic rollback.
- **Isolation:** prohibit colocations that violate tenant, data, or compliance boundaries.

Adoption

Kubernetes exposes GPUs as schedulable custom resources after a device plugin is installed kubernetes.io. That handles device assignment, not sub-model memory admission or fairness. The serving layer must add those controls.

Triton supplies readiness and liveness endpoints plus utilization, throughput, and latency metrics docs.nvidia.com. Kubernetes deployments can control `maxUnavailable` and `maxSurge`, and retain rollout history for rollback kubernetes.io kubernetes.io. These primitives still require an operator to choose safe values and pay for rollout headroom.

Strengths and Limitations

Strengths are lower list cost per physical L4, flexible placement, and the ability to pool idle compute across endpoints. Dynamic batching combines requests and typically raises throughput docs.nvidia.com A pooled scheduler can exploit uncorrelated peaks that twelve isolated VMs cannot share.

Limitations are a larger blast radius, model interaction, host RAM and CPU contention, scheduling complexity, and shared rollout scope. Triton warns that running dynamically allocating models together may exhaust system memory docs.nvidia.com KServe documents that surplus requests wait when container concurrency is exhausted kserve.github.io. Queueing and eviction can therefore create tail-latency coupling even when total memory fits.

General NVIDIA virtual-GPU documentation also says its default best-effort scheduler does not promise a fixed or equal compute share per virtual machine docs.nvidia.com. This is not an AWS G6f benchmark, but it reinforces the need to measure fairness rather than infer it from memory partitions.

Architecture C: CPU Control

Capabilities

A CPU deployment is the control, not an inferior default. Current C7i and C7i-flex processors expose Intel Advanced Matrix Extensions intended to accelerate matrix multiplication for CPU-based machine learning aws.amazon.com Low request volume, compact quantized models, generous latency budgets, or long idle periods can make CPU the least-cost compliant placement.

CPU testing must use the identical artifact-quality threshold, trace, concurrency, warmup, availability policy, and host-cost boundary used for GPU. A cheaper response that misses P95 or changes model quality is not a valid endpoint-hour.

Adoption

CPU fleets use familiar autoscaling and observability controls without a GPU device plugin or GRID driver. Auto Scaling can balance instances across AZs as the group grows docs.aws.amazon.com This makes CPU a useful operational baseline even when it loses the throughput test.

Strengths and Limitations

Strengths include broad regional capacity, simpler images, fine-grained right-sizing, and no stranded accelerator fraction. **Limitations** can include lower throughput, higher latency, more replicas, and worse cost after the SLO forces a large CPU fleet. The test must choose a real candidate instance and price it, not treat “CPU” as free.

The CPU gate is straightforward:

- **Pass quality** at the approved precision and artifact hash.
- **Pass latency** at P95 under solo and mixed trace replay.
- **Pass availability** with the required replica and AZ policy.

- **Pass recovery** during restart and rolling deployment.
- **Compare total cost** only after all four gates pass.

Feature Comparison

Table 1 compares the three architectures at the decision level. Price is only one row because capacity, isolation, and operating model determine how many billable units are required.

Dimension	Per-endpoint G6f	Consolidated whole G6	CPU control
Accelerator profile	Nominal 3 GB, 6 GB, or 12 GB slice; guest table reports 2.79 GiB, 5.59 GiB, or 11.18 GiB docs.aws.amazon.com.	One nominal 24 GB L4; guest table reports 22 GiB.	No accelerator memory; host RAM and vector/matrix support govern fit.
2026 On-Demand compute	g6f.large $0.202/hour; g6f.2xlarge $0.475/hour; g6f.4xlarge $0.950/hour in us-east-1 b0.p.awsstatic.com.	g6.xlarge $0.8048/hour in us-east-1.	Must be priced for the measured CPU shape and replica count.
Fleet rounding	One whole VM per endpoint replica, at least 12 for one replica each.	One whole host per admitted bin, rounded up, plus rollout and failure headroom.	One whole VM per chosen deployment unit, rounded up.
Isolation	Strong endpoint-level VM boundary and small failure domain.	Shared scheduler, memory, CPU, and failure domain unless separated by policy.	Flexible per-endpoint or shared-process isolation.
Software	GRID 18.4 through 19.5 must be validated with the framework and container.	Full-GPU driver path plus multi-model server and admission controls.	CPU runtime, quantization, thread, and instruction-set validation.
Scaling	Independent endpoint autoscaling, but idle capacity is stranded inside each slice.	Pooled demand can improve utilization, but correlated bursts can exhaust the pool.	Independent or pooled, with no accelerator partition constraint.
Best fit	Small validated memory footprint, strict isolation, simple ownership.	Measured mixed workload, compatible isolation, mature serving operations.	Endpoint already passes quality, P95, and availability without GPU.

The table shows why **fractional GPU pricing** is not proportional. A g6f.large one-eighth slice costs roughly 25.1 percent of g6.xlarge, while g6f.4xlarge costs roughly 118.0 percent of g6.xlarge. The larger G6f shape also bundles 64 GiB system memory and one 450 GB NVMe drive, versus 16 GiB and 250 GB for g6.xlarge b0.p.awsstatic.com. Paying more for g6f.4xlarge can still be rational when its host resources or isolation avoid an engineering platform.

Performance and Benchmarks

No public result can establish the winner for twelve private endpoint traces. The reproducible benchmark should replay the same timestamped work against each feasible architecture. MLPerf's server scenario uses a selected queries-per-second rate with Poisson arrivals, an example of why an open-loop arrival process is

more revealing than serial request timing github.com. Microsoft Research also reports that scheduling policies can be fragile under bursty arrivals and GPU memory pressure www.microsoft.com.

The arrival model should be disclosed with the result. Reusing MLPerf's Poisson-style server arrival concept makes the load generator reproducible github.com, while replaying original traces preserves the portfolio's observed correlation. Both are useful, but they answer different questions.

The required test sequence is:

1. **Solo baseline:** run each endpoint alone from cold start through steady state, then capture quality, throughput, P50, P95, errors, CPU, RAM, and peak accelerator memory.
2. **Mixed portfolio:** replay all twelve original traces together while preserving request timestamps and payload distributions.
3. **Correlated burst:** align or amplify observed peaks using a declared transform, then verify queueing, fairness, and errors.
4. **Maximum footprint:** exercise the maximum approved context, batch, concurrency, and output length after warmup.
5. **Model reload:** load, evict, and reload according to policy while other endpoints remain active.
6. **Failure and restart:** terminate one replica or host, measure lost endpoint-minutes, recovery, and overload on survivors.
7. **Rolling deployment:** update one model and the serving layer with the production surge and unavailable limits.
8. **Repeatability:** run enough trials to report distributions and confidence, not one favorable trace.

Dynamic batching should be tuned until the latency budget is approached, not enabled with a universal delay docs.nvidia.com. Continuous batching also changes memory: a larger prefill budget allocates larger input buffers and leaves fewer KV-cache blocks huggingface.co. Memory fill is consequently not compute utilization. Report productive accelerator time or admitted SLO-compliant capacity separately from memory occupancy.

Instrumentation should retain both model-level and host-level views. Triton publishes a used-GPU-memory metric in bytes docs.nvidia.com, and PyTorch exposes the peak tensor allocation counter inside the process docs.pytorch.org. Queue time needs separate capture because KServe's concurrency limit can buffer surplus work kserve.github.io.

Tail results should be presented as percentiles, not only averages, because averages can conceal changes in the tail sre.google. The correlated-burst test deserves its own result because bursty arrivals and memory pressure can expose scheduler fragility www.microsoft.com.

Quality and errors are hard gates. A run that serves more requests by using an unapproved quantization, truncating context, rejecting traffic, or missing P95 is excluded from the cost denominator.

The benchmark packet should be independently reproducible. It should preserve the arrival rule used by MLPerf's server methodology github.com, the percentile rationale for tail latency sre.google, and the exact burst transformation used for stress. It should retain PyTorch's peak memory observation docs.pytorch.org, the vLLM KV-cache allocation docs.vllm.ai, and the batching budget that trades buffers against cache blocks huggingface.co. Those records allow a later price or model change to be evaluated without confusing it with a measurement change.

Data Analysis and Evidence

Dated AWS G6f instance pricing and profile fit

Table 2 combines official us-east-1 Linux shared-tenancy On-Demand prices with the documented guest memory. Monthly figures use **730 metered hours** and exclude EBS, transfer, monitoring, software, and operations. On-Demand Linux instances are billed per second with a 60-second minimum docs.aws.amazon.com.

Instance	GPU allocation	Guest accelerator memory	vCPU / system RAM	Hourly price	730-hour compute
g6f.large	1/8 L4	2.79 GiB	2 / 8 GiB	**$0.2020**	**$147.46** b0.p.awsstatic.com
g6f.xlarge	1/8 L4	2.79 GiB	4 / 16 GiB	**$0.2375**	**$173.38** b0.p.awsstatic.com
g6f.2xlarge	1/4 L4	5.59 GiB	8 / 32 GiB	**$0.4750**	**$346.75** b0.p.awsstatic.com
g6f.4xlarge	1/2 L4	11.18 GiB	16 / 64 GiB	**$0.9500**	**$693.50** b0.p.awsstatic.com
g6.xlarge	1 L4	22 GiB	4 / 16 GiB	**$0.8048**	**$587.50** b0.p.awsstatic.com

The official catalog supports the fractional rates for g6f.xlarge, g6f.2xlarge, and g6f.4xlarge b0.p.awsstatic.com b0.p.awsstatic.com b0.p.awsstatic.com. The table makes the surprising result explicit: the half-L4 g6f.4xlarge costs more per hour than the whole-L4 g6.xlarge because the instance bundles substantially more host capacity.

Twelve-endpoint break-even

Assume only for the arithmetic that each 3 GB endpoint fits g6f.large host resources, every endpoint needs one always-on replica, and all twelve pass the same SLO gates. The isolated fractional fleet is:

`4 × \$0.202 + 4 × \$0.475 + 4 × \$0.950 = \$6.508/hour.`

At **$4,750.84/month**, that fleet costs the same as 8.09 g6.xlarge hosts, but billable fleet size is an integer. Therefore:

- **Four whole hosts:** $3.2192/hour and $2,350.02/month, 50.5 percent below the fractional compute floor b0.p.awsstatic.com.
- **Eight whole hosts:** $6.4384/hour and $4,700.03/month, still $50.81/month below the fractional floor b0.p.awsstatic.com.
- **Nine whole hosts:** $7.2432/hour and $5,287.54/month, now $536.70/month above the fractional floor b0.p.awsstatic.com.

This is the **AWS G6f break-even analysis**: the consolidated one-replica architecture wins on compute list price only while it needs eight or fewer g6.xlarge hosts. It does not prove that four hosts are feasible. The documented profile ceilings total 78.24 GiB guest-visible memory across the portfolio. Four 22 GiB hosts provide 88 GiB, and one endpoint from each size class totals 19.56 GiB per host. That leaves 2.44 GiB per host before measured reserve, runtime variation, rollout, and failures docs.aws.amazon.com. Any isolation rule can invalidate that neat packing.

For two replicas per endpoint, G6f begins at **24 instances and $9,501.68/month**. A whole-GPU design needs separate copies across failure domains, so duplicating the four-host placement begins at eight hosts and $4,700.03/month b0.p.awsstatic.com. It may need more hosts for N+1 recovery, correlated peaks, or rollout surge. Auto Scaling's AZ balancing and replacement behavior help implement the policy, but they do not remove the spare-capacity bill docs.aws.amazon.com.

Sensitivity and complete cost

Table 3 is a decision grid, not a benchmark. It shows which input should trigger a recalculation and the direction of pressure.

Variable	Lower-risk condition	Higher-risk condition	Capacity-plan effect
Peak correlation	Peaks are demonstrably independent.	Several traces peak together.	Pooling benefit shrinks; whole-GPU hosts or queues increase.
Replica minimum	One replica is permitted.	Two or more replicas across AZs.	G6f instance count multiplies; G6 also needs duplicated placements and spares.
Memory growth	Peak plus reserve remains within current bin.	New context, batch, runtime, or weights cross a ceiling.	Move profile, repack, or add host before deployment.
Packing count	Four mixed models fit with measured reserve.	Isolation or resource coupling forces fewer models per host.	Whole fleet approaches the nine-host compute break-even.
Commitment	Stable baseline can use an eligible Savings Plan.	Demand is uncertain or short-lived.	Compare like-for-like effective rates, never mix commitment and On-Demand.
Failure policy	Spare capacity can absorb one host loss.	Each AZ must survive a host loss plus peak.	Add integer hosts and retest overload behavior.
Engineering burden	Existing measured multi-model platform.	New scheduler, dashboards, runbooks, and on-call skills.	Report engineer-hours separately, or multiply by an approved loaded rate.

The full monthly equation is **compute + storage + transfer + monitoring + verified software or licenses + buyer-valued operations**. AWS's gp3 example uses $0.08 per GB-month and includes 3,000 input/output operations per second plus 125 MB/s baseline throughput aws.amazon.com aws.amazon.com. CloudWatch's US East example prices the first 10,000 metrics at $0.30 each, while its log example uses $0.50 per ingested GB aws.amazon.com aws.amazon.com. Those are examples, so the buyer should pull current regional charges and measured volumes.

Commitments do not repair poor placement. Compute Savings Plans can apply across EC2 instance families, while EC2 Instance Savings Plans advertise up to 72 percent off On-Demand in exchange for a regional family commitment docs.aws.amazon.com docs.aws.amazon.com. Spot should be modeled only for interruption-tolerant replicas because a rebalance recommendation can arrive with the two-minute interruption notice docs.aws.amazon.com.

The reporting unit should be:

`SLO-compliant endpoint-hour cost = total metered fleet cost / endpoint-hours passing every quality, latency, availability, and error gate`.

FinOps guidance recommends unit costs rather than total cost alone for strategic decisions www.finops.org. This denominator prevents a cheap but unavailable endpoint from looking efficient.

The same unit-economic principle should be applied separately to each endpoint class, not only to the aggregate portfolio www.finops.org. Otherwise, one high-volume endpoint can hide an expensive idle placement elsewhere.

Any complete cost workbook should keep its unit-cost definition beside the results www.finops.org. That makes the denominator reviewable when availability policy changes.

Implications and Future Directions

The practical output is a **placement rule**:

- **Choose CPU** when the measured CPU candidate passes identical quality, P95, availability, and recovery gates at lower complete cost.
- **Choose G6f** when the smallest validated profile has adequate guest-visible memory, CPU, RAM, network, storage, driver compatibility, and reserve, especially when endpoint isolation or independent change control is valuable.
- **Choose whole G6** when the mixed portfolio passes under correlated peaks, failure, reload, and rolling deployment, and its integer host fleet plus engineering cost remains below isolated alternatives.
- **Use a hybrid** when only some endpoints benefit from pooling. ECS can represent both fractional and full GPU capacity, while the placement policy retains explicit endpoint classes.
- **Reject untested fit** when only weight-file size, nominal profile labels, or average latency is available.

Whole-GPU packing efficiency should be defined as measured productive accelerator time or admitted SLO-compliant workload capacity divided by provisioned whole-L4 capacity. It must not be inferred from memory occupancy. NVIDIA's virtual-GPU documentation states that assigned framebuffer remains reserved for a vGPU's lifetime docs.nvidia.com. It also describes time-sliced streaming multiprocessors and engines as serving one vGPU at a time in scheduled slices docs.nvidia.com. AWS-specific performance must still be measured rather than inferred from this general architecture.

Remeasure when any of these changes:

- **Artifact:** weights, precision, runtime, CUDA, driver, framework, or serving-engine version.
- **Demand:** request rate, payload, context, batch, concurrency, or peak correlation.
- **Service policy:** P95 target, quality threshold, replica minimum, AZ distribution, or isolation class.
- **Economics:** regional rate, commitment coverage, storage, transfer, monitoring, license, or loaded engineering rate.
- **Operations:** startup, reload, failure recovery, rollout surge, or on-call burden.

Operational remeasurement should include rollout configuration because Kubernetes exposes explicit surge and unavailable controls kubernetes.io. It should also retain repository and model-load state because Triton's explicit mode requires operator-initiated load and unload after startup docs.nvidia.com.

The change test should preserve Kubernetes rollout history kubernetes.io, record buffered requests when concurrency is exhausted kserve.github.io, and rerun the correlated-burst case that challenges scheduling under memory pressure www.microsoft.com. Hardware documentation should travel with the record too, including the L4's physical 24 GB capacity www.nvidia.com, so nominal and guest-visible units are not silently conflated.

GPU Smith describes optimization work spanning quantization, batching, KV cache, and scheduler tuning gpusmith.com. For an independent advisor, the appropriate role is to make these assumptions reproducible and disclose the acceptance boundary, not to appear as another row beside AWS instances.

Its first-party description also says engagements use written acceptance criteria and an as-built documentation set gpusmith.com. In this comparison, that approach translates to retaining the inventory, traces, configuration, evidence, and signed placement rule together.

Frequently Asked Questions (FAQs)

Is G6f the cheapest AWS GPU for small inference endpoints?

Not universally. **g6f.large at $0.202/hour** is the lowest-priced accelerator shape in this comparison, but twelve separately rounded endpoint replicas can cost more than a smaller consolidated whole-L4 fleet. CPU can be cheaper still when it meets the same SLO. Host-resource fit, replicas, idle minimums, and isolation determine the answer.

Does a 3 GB model fit the 3 GB G6f profile?

The label alone is insufficient. AWS's detailed table reports 2.79 GiB to the guest for the one-eighth profile. Measure post-warmup maximum allocation under approved context, batch, and concurrency, then subtract a declared reserve. TensorRT workspace and KV cache make weight-file size an incomplete capacity measure.

The measurement can combine a framework peak counter docs.pytorch.org with an explicit KV-cache allocation docs.vllm.ai and serving-level GPU memory telemetry docs.nvidia.com. The maximum observed across the approved test matrix is the input to admission, not the artifact's file size.

Is a fractional L4 slower than a full L4?

It has a smaller allocated share, but no universal workload penalty should be invented. Framework compatibility, time slicing, concurrency, batch efficiency, CPU and memory coupling, and trace shape all matter. Compare SLO-compliant throughput and P95 with the same artifact and trace.

When does whole G6 break even for this twelve-endpoint portfolio?

At the specified us-east-1 On-Demand rates, the isolated one-replica G6f floor is $6.508/hour. Eight g6.xlarge hosts cost $6.4384/hour, while nine cost $7.2432/hour. Thus the list-price break-even lies between eight and nine whole hosts, before ancillary cost and engineering burden.

These values derive from the official g6.xlarge catalog rate b0.p.awsstatic.com and the three fractional catalog rates shown in Table 2.

How should high availability change the calculation?

Use integer replicas and explicit AZ placement. Duplicate placements across failure domains, reserve enough capacity for a host loss, and test survivor overload. A statement such as “two replicas” is incomplete unless it identifies whether both can run at peak after one AZ or host becomes unavailable.

Conclusion

For the defined portfolio, **consolidated whole-L4 G6 has the lower compute floor**, but only under a validated packing and operating model. Four mixed g6.xlarge hosts cost about $2,350 per 730-hour month versus about $4,751 for twelve isolated G6f replicas. The advantage survives through eight whole hosts and reverses at nine b0.p.awsstatic.com. That arithmetic answers the narrow break-even question, not the placement question.

Per-endpoint G6f remains the clearer choice when isolation, independent deployment, small blast radius, or host-resource needs justify its premium. CPU remains the correct control wherever it meets quality, P95 latency, recovery, and availability. Whole G6 is appropriate when measured demand is sufficiently complementary and the buyer can operate memory admission, fairness, health, telemetry, rollback, and failure headroom.

The durable decision is therefore conditional: inventory every endpoint, measure peak runtime memory and host demand, replay identical traces, enforce quality and SLO gates, round every fleet to whole instances, add all ancillary and operational cost, and repeat after material change. Fractional GPU inference cost is a fleet

outcome, not a GPU fraction multiplied by a list price.

That conclusion follows the same unit-cost discipline recommended by the FinOps Foundation www.finops.org: compare only endpoint-hours that actually satisfy the service objective.

Source links

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

1. https://b0.p.awsstatic.com/pricing/2.0/meteredUnitMaps/ec2/USD/current/ec2-ondemand-without-sec-sel/US%20East%20%28N.%20Virginia%29/Linux/index.json
2. https://docs.aws.amazon.com/ec2/latest/instancetypes/ac.html
3. https://aws.amazon.com/ec2/instance-types/g6/
4. https://docs.nvidia.com/deeplearning/tensorrt/latest/reference/troubleshooting-faq.html
5. https://docs.vllm.ai/en/latest/api/vllm/config/cache/
6. https://docs.nvidia.com/deeplearning/triton-inference-server/user-guide/docs/user_guide/optimization.html
7. https://aws.amazon.com/about-aws/whats-new/2025/07/amazon-ec2-g6f-instances-fractional-gpus/
8. https://www.nvidia.com/content/dam/en-zz/Solutions/Data-Center/l4/PB-11316-001_v01.pdf
9. https://sre.google/sre-book/service-level-objectives/
10. https://gpusmith.com/
11. https://docs.aws.amazon.com/AmazonECS/latest/developerguide/ecs-gpu-specifying.html
12. https://docs.pytorch.org/docs/main/notes/cuda.html
13. https://huggingface.co/docs/transformers/en/continuous_batching
14. https://docs.aws.amazon.com/ec2/latest/instancetypes/ec2-instance-regions.html
15. https://docs.aws.amazon.com/ec2/latest/instancetypes/ec2-instance-quotas.html
16. https://docs.aws.amazon.com/autoscaling/ec2/userguide/what-is-amazon-ec2-auto-scaling.html
17. https://docs.aws.amazon.com/en_eu/AWSEC2/latest/UserGuide/nvidia-GRID-driver.html
18. https://docs.nvidia.com/cuda/vGPU/
19. https://docs.nvidia.com/datacenter/cloud-native/container-toolkit/latest/docker-specialized.html
20. https://docs.nvidia.com/deeplearning/triton-inference-server/user-guide/docs/index.html
21. https://docs.nvidia.com/deeplearning/triton-inference-server/user-guide/docs/user_guide/model_management.html
22. https://kubernetes.io/docs/tasks/manage-gpus/scheduling-gpus/
23. https://kubernetes.io/docs/concepts/workloads/controllers/deployment/
24. https://docs.nvidia.com/deeplearning/triton-inference-server/user-guide/docs/user_guide/batcher.html
25. https://docs.nvidia.com/deeplearning/triton-inference-server/archives/triton-inference-server-2360/user-guide/docs/user_guide/rate_limiter.html
26. https://kserve.github.io/archive/0.12/modelserving/autoscaling/autoscaling/
27. https://docs.nvidia.com/knowledge-base/latest/vgpu-features.html
28. https://aws.amazon.com/ec2/instance-types/c7i/
29. https://github.com/mlcommons/inference_policies/blob/master/inference_rules.adoc

30. https://www.microsoft.com/en-us/research/publication/beyond-prediction-tail-aware-scheduling-for-llm-inference/
31. https://gpusmith.com/articles/en/good-gpu-utilization-percentage
32. https://docs.nvidia.com/deeplearning/triton-inference-server/archives/triton-inference-server-2320/user-guide/docs/user_guide/metrics.html
33. https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-on-demand-instances.html
34. https://aws.amazon.com/ebs/pricing/
35. https://aws.amazon.com/cloudwatch/pricing/
36. https://docs.aws.amazon.com/savingsplans/latest/userguide/plan-types.html
37. https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/rebalance-recommendations.html
38. https://www.finops.org/wg/introduction-cloud-unit-economics/
39. https://docs.nvidia.com/ai-enterprise/release-7/latest/infra-software/vgpu/overview.html
40. https://gpusmith.com/articles/en/llm-inference-hardware-sizing-guide

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