

SAMPLE AUDIT EXCERPT

AI / ML estate with self-hosted training and SageMaker inference

Simulated example. Not a client engagement. Built on a constructed AWS environment so a buyer can see the report format. Every figure below is computed from the stated inputs: annualized × 365/30 for the 30-day baseline month, and each bound of a range rounded down to the nearest \$100. None is measured from a live account.

Prepared by Xavier Perera, Founder · Format date: October 2026 · Baseline convention in this sample: June 2026 (30 days).

1. Environment assumptions

Item	Simulated value
Workload	Self-hosted GPU training on EC2 (p4d / g5 class) + SageMaker real-time inference + supporting API
Region	us-east-1
Monthly net AWS spend (baseline month)	\$68,000
Cost metric	Net amortized cost, credits / refunds / tax excluded
Training purchase option	100% on-demand; 0% Spot; 0% Savings Plan
Inference	SageMaker real-time endpoints on ml.g5-class, 24/7
Pricing basis	Public us-east-1 list rates (AWS Price List, checked October 2026) unless noted
Access model	Read-only. No production changes. No application or dataset contents are read.

Hard scope limits for this workload

- GPU utilization is not in CloudWatch by default. A 5-day read-only audit can claim idle GPU hours only if NVIDIA metrics already exist (CloudWatch agent nvidia_gpu, SageMaker DCGM / detailed observability). If they do not, idle-GPU is a hypothesis with a measurement plan. It is not a dollar finding.
- Compute Savings Plans do not cover SageMaker. SageMaker training and endpoints need a SageMaker Savings Plan. EC2 training nodes need Compute or EC2 Instance Savings Plans. These are separate plan types.
- SageMaker Serverless Inference does not support GPUs. It is not a fallback for ml.g5 endpoints.

2. Executive summary

Opportunity	Monthly range	Annualized range	Notes
Stop / terminate training instances after the job ends	\$2,300-\$3,900	\$28,500-\$47,500	Only if GPU or instance-idle evidence already exists
Checkpoint / artifact retention on S3	\$800-\$1,300	\$10,800-\$16,600	Storage metrics + lifecycle. No object-content access required
SageMaker endpoint scale policy (not Serverless GPU)	\$0-\$2,600	\$0-\$31,600	Invocation metrics; latency SLO is the constraint
Excerpt total (non-overlapping)	\$3,200-\$7,800	\$39,300-\$95,800	Lower bound is the number counted
1-year reserved GPU / Savings Plan before the floor is clean	\$0 claimed	\$0	Evaluated. Recommended against.

The ranges above are what the constructed estate can defend. Annualized × 365/30 for the 30-day baseline month. Each bound is rounded down to the nearest \$100 from unrounded figures, so a total can be up to \$100 per row above the sum of its rounded rows. This sample is illustrative and is not covered by the guarantee. The guarantee applies only to live engagements, subject to the terms of the engagement agreement.

FINDING A · COMPUTE Training instances that keep billing after the job ends

Mechanism. On-demand GPU instances bill while running, including nights and weekends between experiments. The expensive habit is leaving the box up. The cheap fix is terminate-on-completion in the training pipeline, with a utilization alarm as backstop.

Evidence a live audit would use (in order)

1. EC2 state + instance hours from CUR, grouped by family (p4d, g5, etc.) and purchase option.
2. Launch and stop times from ec2:DescribeInstances and resource-level hours in the billing export; job start and end times from orchestrator records your team shares. SageMaker training cost, if any, comes from billing data and CloudWatch SageMaker metrics.
3. Only if present: nvidia_smi_utilization_gpu or DCGM GPU util overlaid on those hours.
4. Compute Optimizer idle recommendations for G/P ASGs, if enabled: a useful cross-check, not sufficient alone.

If step 3 is missing, Valmora reports "GPU-idle cannot be measured from current telemetry" and does not put a utilization % in the executive summary.

Simulated inputs (constructed)

GPU training compute in baseline month: \$28,400. Share of instance-hours with near-zero CPU and (where metrics exist) GPU util <5%: constructed at 15–25% of that line. $15\text{--}25\% \times \$28,400 = \$4,260\text{--}\$7,100$ if applied to the whole training line. We claim only the idle hours that fall between jobs, constructed at 55% of idle hours, not low-util hours during a running job (low GPU util during training is a model/batch-size problem, not a stop-instance problem). $55\% \times \$4,260 = \$2,343$ to $55\% \times \$7,100 = \$3,905$. Claimed range, rounded down: \$2,300–\$3,900 / month; annualized $\times 365/30$: \$28,500–\$47,500.

Assumptions	Jobs are finite and orchestrated. Interactive notebooks left up on purpose are out of this finding. A 60-minute idle threshold does not kill back-to-back experiments. Stopping vs terminating: stop keeps EBS; terminate does not. The cheaper action depends on whether the root volume is disposable.
Effort / risk	Effort: low if the orchestrator already has a completion hook; medium if training is launched by hand. Risk: low if the stop is tied to job completion. Risk rises if a shared instance hosts more than one job.
Next action	Tag training instances with job ID at launch. Terminate (or stop) in the job-completion step. Add a CloudWatch alarm on GPU util only if the agent is already publishing it; do not install agents inside the audit.
Validation / rollback	Job success rate and time-to-start of the next run. Rollback = disable the completion hook.
Overlap rule	Do this before any Spot fleet or Savings Plan. Commitments sized against idle hours lock the waste in.

FINDING B · STORAGE**Checkpoint and artifact sprawl on S3**

Mechanism. Training writes a checkpoint every N steps. Nothing expires. S3 Standard bills the pile forever. This is a retention-policy problem, not a “last-access lifecycle Days” problem.

Important AWS constraint. S3 Lifecycle Days is days since object creation / last modified, not days since last access. Access-based movement is S3 Intelligent-Tiering (or a manual process informed by Storage Class Analysis). A rule written as “transition if unused 90 days” via standard Lifecycle Days will move young, still-hot objects that simply have an old-looking prefix if you key off the wrong metadata.

Evidence a live audit would use

- S3 bytes and object counts by bucket and storage class from billing data and CloudWatch storage metrics. Prefix-level age needs an inventory report your team shares.
- GetBucketLifecycle: presence or absence of expiration / transition rules on checkpoint prefixes.
- No s3:GetObject on training data or weights.

Simulated inputs (constructed)

Checkpoint / artifact prefixes: 60,000 GB (60 TB) in S3 Standard. Public tiered rate, assuming these prefixes are the account's only S3 Standard storage: \$0.023 / GB-month for the first 51,200 GB and \$0.022 above, so \$1,371.20 / month if all of it were deletable. Usually some is kept. Retain: best-metric checkpoint + final checkpoint per run + registered model artifacts. Constructed deletable / immediately-expirable share: 65–100% of that prefix after the ML team sets the rule (100% only where the checkpoints to keep and the registered artifacts are first moved out of it). At 65%, the remaining 21,000 GB cost \$483.00, so the saving is \$888.20; at 100% it is \$1,371.20. Claimed range, rounded down: \$800–\$1,300 / month; annualized $\times 365/30$: \$10,800–\$16,600.

Assumptions	The ML team will accept “keep best + final per run; expire intermediates at 30 days.” Registered model artifacts are archived, not deleted. Retrieval from Glacier Instant Retrieval is rare.
Effort / risk	Effort: low once the retention rule is written down. Risk: low if expiration is limited to intermediates. High if someone points a lifecycle expiration at the model registry prefix.
Next action	Agree the retention rule with the ML owner. Apply lifecycle expiration on intermediate checkpoint prefixes. Move registered artifacts to Glacier Instant Retrieval or a dedicated bucket with no expiry. Use Intelligent-Tiering on prefixes with unknown access, not a creation-date transition that pretends to be last-access.
Validation / rollback	Lifecycle expiration is not reversible. Test on one prefix first. Object count should flatten, then fall.

FINDING C · COMPUTE

SageMaker real-time endpoints sized for a launch-day peak

Mechanism. Real-time endpoints bill the instance while the endpoint exists. If 90-day p95 invocations use a fraction of provisioned copies and no scaling policy exists, the estate is paying for a peak that does not arrive.

What this is not. Not SageMaker Serverless Inference. Serverless does not support GPUs and maxes at 6 GB memory. It is the wrong recommendation for ml.g5 endpoints. Not Compute Optimizer "Idle SageMaker endpoint," which fires on zero invocations over 14 days. An endpoint with sparse traffic is over-provisioned, not idle.

Evidence a live audit would use

- SageMaker training and endpoint cost from billing data, and CloudWatch SageMaker metrics (Invocations, InvocationsPerInstance, ModelLatency, OverheadLatency) for 30–90 days. Endpoint configuration is confirmed with your team.
- Compute Optimizer idle recs only as a filter for endpoints with zero invocations.

Simulated inputs (constructed)

Production endpoints: three ml.g5-class copies, 24/7. 90-day p95 load constructed at roughly one-third of provisioned copies. No target-tracking policy attached. Endpoint cost in the baseline month (constructed): \$7,800. A clean one-third cut with no SLO hit removes one-third of that: $\$7,800 \times 1/3 = \$2,600$ / month. That is the upper bound, not the expected case.

Economics	Claimed range: \$0–\$2,600 / month; annualized $\times 365/30$: \$0–\$31,600. The upper bound assumes stepping down one instance at a time with latency alarms armed. The lower bound is \$0 because if p95 latency is already near SLO, no instance can come out until the model is distilled or moved to async inference for non-interactive traffic.
Assumptions	Cold-start / scale-out time is acceptable for the product. Burst traffic is real but rare; target-tracking can cover it. Instance-class downsize (g5.xlarge with a smaller GPU, or Inferentia) is a separate project and is not in this number.
Effort / risk	Effort: medium. Risk: medium. Inference latency SLO. Mitigation: one endpoint, one instance, one week.
Next action	Attach target-tracking on InvocationsPerInstance. Reduce desired count toward p95 + one replica of headroom. Do not migrate these endpoints to Serverless.
Validation / rollback	p99 ModelLatency and error rate vs the prior week at comparable traffic. Rollback = restore the previous instance count.

Rejected: 1-year GPU commitment against today's training line

Evaluated. Not claimed. A 1-year SageMaker Savings Plan (if the fleet is SageMaker) or Compute / EC2 Instance Savings Plan (if the fleet is EC2) would discount a large on-demand GPU line. Two reasons it is the wrong action now: (1) Finding A has not been banked. A commitment sized against idle hours locks in that cost. (2) Training cadence and GPU family are still moving. Savings Plans are not cancellable after a narrow 7-day / same-calendar-month / $\leq \$100$ -hour return window. Unused commitment is spent.

Spot on checkpointed, resumable training remains a candidate after idle hours are gone and interruption handling is proven. Published Spot discounts of 60–70% are a rate-card ceiling, not a capacity guarantee on p4d. That work is sequenced later and is not in the excerpt total. Cheap GPU hours you cannot unwind are not cheap.

Sequencing

1. Measure GPU idle properly if telemetry is missing, or claim only instance-hours between jobs from CUR + orchestrator records your team shares.
2. Finding A (stop after job) and Finding B (checkpoint expiry): week 1.
3. Finding C (endpoint policy): month 1, SLO-gated.
4. Spot on interruptible training: after checkpoints resume cleanly and data is local to the training region.
5. Savings Plan: after a 60-day post-cleanup baseline, against the residual always-on floor only, on the correct plan family (Compute vs SageMaker).

Methodology, scope, guarantee

Same commercial rules as the SaaS excerpt: completed billing month baseline; annualize $\times 365$ / days; net amortized cost; credits / refunds / tax out; recurring only; no double count; lower bound of a range is what identification accounting may count.

Read-only describe + billing + metrics. No s3:GetObject on datasets or weights. No endpoint invocation. No agent install. If GPU metrics are absent, the report says so and does not estimate utilization.

Guarantee. This sample is illustrative and is not covered by the guarantee. The guarantee applies only to live engagements, subject to the terms of the engagement agreement.

Xavier Perera, Founder, Valmora Consulting · xavier@valmoraconsulting.com · valmoraconsulting.com