

AWS COST · SAMPLE DELIVERABLE

The Analysis: Waste Register, Rightsizing, Commitments, Savings Roadmap

Week 2 of 6: Find and rank · prepared for Tidepost, Inc.

Prepared by Charles Sieg · charles.sieg@vantalect.com · vantalect.com

April 16, 2026 · AWS Cost Analysis & Optimization: The Optimization (six weeks)

SAMPLE DELIVERABLE

This is a sample deliverable. Tidepost, Inc. is a fictional company: its people, products, users, accounts, investors, and figures are invented for this sample, and any resemblance to a real company, person, product, or publication is coincidental. AWS prices are list prices from the AWS Price List API retrieved on October 13, 2026, and from the public AWS Price List bulk API on October 18, 2026.

Contents

- Summary 2
- How the changes were priced 5
- The waste register 7
- Compute and machine-learning sizing 13
- Databases, caches, and the telemetry stream 17
- Moderation and language 20
- Storage and the data lake 22
- Logging and metrics 25
- Networking and delivery 26
- Commitments 28
- The savings roadmap 29
- What is not counted 37

Summary

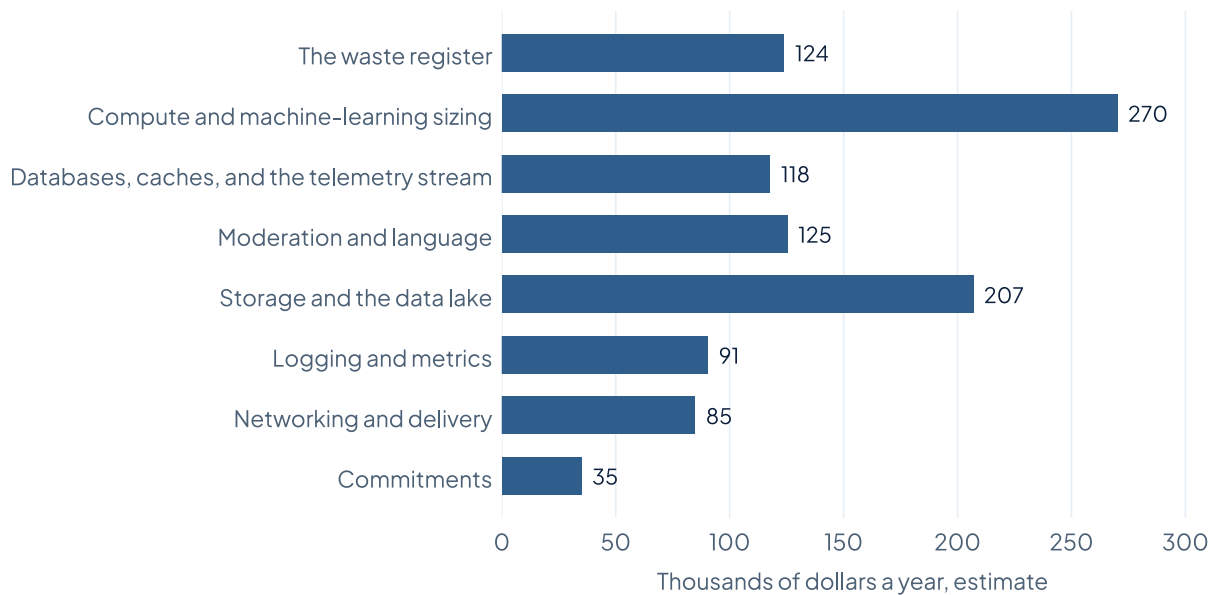
The Analysis finds 37 problems in Tidepost’s AWS bill and prices them as 40 changes worth \$1,054,404 a year at the estimate, 50.0% of the \$2,110,762 April run rate. Dev Raman and Kofi Brandvold: this is the second report of the cost engagement. Every change has an owner, an account, evidence from Tidepost’s own data, a price from the AWS Price List, and a prescriptive change that Tidepost’s engineers can make through a pull request. Every figure is an estimate. Nothing has been decided: the sponsor readout is tomorrow, April 17, the owners decide change by change, and week six measures each change that was made on Tidepost’s Cost and Usage Report.

What you most need to know:

- 1. 40 changes, \$87,867 a month, \$1,054,404 a year.** 37 findings priced as 40 changes, because three findings split into two changes where the mechanism differs: the embedding endpoint's instance size and its scaling (SM-1a and SM-1b), the staging rebuild and its schedule (EC2-1a and EC2-1b), and the delivery ladder and the transcoding it removes (CDN-1 and CDN-1b). The changes are in eight sections ([The Analysis by section, at estimate. Each year figure is the sum of the changes as printed, each rounded half up; the monthly figures are the unrounded sums, so a row's monthly figure times twelve can differ from its yearly figure by a dollar](#)).
- 2. The ten largest changes are \$606,100 a year, 57.5% of the total.** S3-2 (Originals over 90 days to Glacier Instant Retrieval, \$120,787), SM-1a (Embedding endpoint to ml.g5.2xlarge, \$97,674), and MOD-1 (Rekognition Video for new accounts and reported videos only, \$83,160) are the three largest ([The savings roadmap](#)).
- 3. The waste register is seven changes and \$123,958 a year, and none needs a new design.** Notebooks that never stop, a Kinesis stream that keeps a year of data nobody reads, and snapshots and volumes nothing uses ([The waste register](#)).
- 4. Five changes wait on a condition outside the week:** SM-3, EC2-1a, EC2-1b, S3-2, CACHE-2. Each is named with what it waits for, and the owner sets its date at the readout.
- 5. Commitments are priced last, on what is left.** The Compute Savings Plan (SP-1) and the SageMaker Savings Plan (SP-2) are sized on the always-on usage that remains after every other change, so a dollar is never saved twice ([Commitments](#)).
- 6. Five Well-Architected HRIs close with these changes.** PERF03, COST06, COST07, SUS02, and SUS04 were accepted in writing in March until this engagement; [Which change closes which HRI](#) shows which change closes which.

The Analysis by section, at estimate. Each year figure is the sum of the changes as printed, each rounded half up; the monthly figures are the unrounded sums, so a row's monthly figure times twelve can differ from its yearly figure by a dollar.

Section	Changes	A month	A year	Share of the total	Share of the April bill
The waste register	7	\$10,330	\$123,958	11.8%	5.9%
Compute and machine-learning sizing	10	\$22,511	\$270,134	25.6%	12.8%
Databases, caches, and the telemetry stream	7	\$9,815	\$117,778	11.2%	5.6%
Moderation and language	3	\$10,444	\$125,322	11.9%	5.9%
Storage and the data lake	5	\$17,241	\$206,898	19.6%	9.8%
Logging and metrics	3	\$7,543	\$90,517	8.6%	4.3%
Networking and delivery	3	\$7,070	\$84,841	8.0%	4.0%
Commitments	2	\$2,913	\$34,956	3.3%	1.7%
All sections	40	\$87,867	\$1,054,404	100.0%	50.0%



Estimated savings by section, in thousands of dollars a year.

This is an estimate, not a result

The Analysis prices what a change should save from quantities and unit prices. It does not say what a change will save. Week six reads the Cost and Usage Report before and after each change and reports what happened, change by change, including every miss and the physical reason for it. Until then, every number here is a price list multiplied by a quantity.

How the changes were priced

Every change is a before and an after for exactly what it touches, priced from a quantity and a unit price. The quantities come from the April run rate in the baseline and from the evidence named on each change; the unit prices come from the AWS Price List for us-east-1, retrieved on the dates in the notice at the front of this report. A change's saving is the monthly cost of what it touches before, less the monthly cost after, times twelve. Months are 730 hours.

The prices

6 of the 40 changes (MOD-1, MOD-2, TR-1, SM-1a, SM-1b, KIN-1) are priced from the public AWS Price List bulk file, because the first table had no price for Rekognition moderation, Transcribe, SageMaker asynchronous inference on the m1.g5.12xlarge, or Kinesis long-term retention. Those unit prices are used exactly as the file gives them, and the changes that use them say so. The same file prices 22.3% of the April bill, as the baseline shows. Every other change uses the first table. Nothing is estimated from a price AWS does not publish.

S3 savings come off the Standard tier they really leave. Tidepost stores 1,087 TB in Standard, which bills \$0.021 a GB-month above its first 500 TB, so the first 587 TB to leave come off that top tier. The S3 changes are priced in order (S3-4, S3-3, S3-1, then S3-2): the first three take 325 TB, all at \$0.021; S3-2 takes the last 262 TB of the top tier, and its other 301 TB come off the \$0.022 tier.

Rounding. Each figure is rounded half up, one row at a time, and a total is the sum of the rows as printed. Where a table's rows are rounded and its total is rounded from the unrounded sum, the caption says so.

Interactions, in order

When two changes touch the same usage, the second is priced on what the first leaves, so a dollar is counted once. This is the order the Analysis prices in, and the reason for each.

Pricing order. A change in the first column is never also counted in the second.

Priced second	After	Why the order matters
MOD-2: Frame sampling every two seconds	MOD-1	Halving the sampled frames saves only on the frames that MOD-1 still leaves for the frame path
SM-1b: Embedding endpoint scales on its backlog	SM-1a	Scaling on backlog saves on the smaller instance that SM-1a leaves, not on the instance it replaces
EC2-4: API tier on Graviton	EC2-3	Graviton saves on the average number of API instances that EC2-3 leaves, not on the old fixed count
CDN-1b: One rendition fewer to transcode	EC2-5	Skipping a rendition saves a share of the transcoder hours that EC2-5 leaves
LOG-2: Application logs kept 30 days	LOG-1	Shorter retention saves on the volume that LOG-1 leaves, which is smaller
ATH-1: Athena on partitioned Parquet	S3-1	Partitioned Parquet saves on the data S3-1 produces, and S3-1 moves the raw JSON out of Athena's reach
CACHE-2: Redis OSS to Valkey	CACHE-1	Valkey saves on the cache nodes that CACHE-1 leaves, which are half the price
SP-1 and SP-2	Every other change	The commitments are sized on the floor of always-on usage that every in-place and waiting change leaves

How a change gets made

Tidepost's engineers make every change, through a pull request to `tidepost-cdk` that the service owner reviews and Tidepost's pipeline ships. We pair on the pull request and review it. We never write to production: the access role cannot, and the Analysis is written for that. Five rules apply to every change in the sections that follow.

- 1. The owner signs off first.** Nothing is deleted or resized on our word alone.
- 2. A change has a way back,** and the card says what it is. Where there is no way back, the card says so and names what is checked first.
- 3. The change goes in code.** A console change is undone by the next deployment, so the pull request changes the CDK, not the console.
- 4. The bill proves it.** Week six reads the Cost and Usage Report before and after each change.

5. **Content stays unread.** The role that reads for the Analysis cannot read the contents of a bucket, table, secret, or log message, so nothing here depends on content, and where a number needs a count from a log, a Tidepost engineer runs it.

How week six will measure each change

This report prints no measured figure, and neither do the next five. This is how week six measures, so that the Analysis and the measurement are judged by the same rule.

- **Every change except the commitments:** the public On-Demand cost of the resources the change touched, on the Cost and Usage Report's `pricing_public_on_demand_cost` column (the cost with no Savings Plan or discount applied), for the seven days before against the seven days after, with the change day excluded, annualized. Using that column keeps SP-1's discount from being counted again in any other change.
- **SP-1:** by its Savings Plan ARN, the On-Demand cost of the usage it covered, less its fee.
- **S3 changes:** by prefix and storage class, from S3 Inventory and S3 Storage Lens, because the telemetry, originals, and renditions changes land on the same buckets in the same weeks.
- **CDN-1:** normalized to bytes per view, so that a rise in traffic is not counted as a saving.
- **Business Support+:** billed monthly and tagged with nothing, it appears in no daily series and in no comparison.

The Cost and Usage Report is read through May 13, because its daily rows finalize about 48 hours after the day ends. A change made after May 6 would have fewer than seven full days after it, so week six cannot measure it. The changes that wait on a condition may land after that date for that reason.

The waste register

\$123,958 a year (7 changes) pays for resources nobody uses or for capacity that sits idle.

Removing it needs no new design, only an owner's yes and a safe way back. The register at the end of this section ([The waste register at estimate](#)) lists the seven; the savings roadmap ranks them against the other 33.

The seven changes

SM-2: Notebook auto-stop

Owner: Yuki Sorensen, Head of Machine Learning. **Account:** tp-sandbox-ml. **Proposed for:** week 3 (April 20 to 24). **Saves:** \$61,266 a year (\$5,106 a month), rank 4 of 40. **Status:** proposed. **Well-Architected:** one of the three changes that close COST06, which closes when SM-1a lands; one of the two changes that close SUS02, which closes when SM-1a lands; accepted in writing until this engagement.

What we saw. Kernel activity: the notebooks were idle 74% of their hours.

The change. Attach a lifecycle configuration that stops each notebook instance after 60 idle minutes. The kernel-activity evidence shows the notebooks idle 74% of their hours, so the stop costs a data scientist a restart, not a lost run. A notebook has to be stopped before a configuration can be attached, so the owners do it notebook by notebook at the end of a working day. A notebook running a long training job carries the tag `autostop-exempt` and keeps running. The same configuration goes into the sandbox account's template, so a notebook created later starts with it. **Rollback:** detach the configuration and start the notebook.

BASH

```
# autostop-on-start.sh stops the instance after 60 idle minutes unless it carries the autostop-exempt tag.
aws sagemaker create-notebook-instance-lifecycle-config --notebook-instance-lifecycle-config-name autostop-60 \
  --on-start Content="$(base64 < autostop-on-start.sh | tr -d '\n')"
aws sagemaker stop-notebook-instance --notebook-instance-name <notebook-name>
aws sagemaker wait notebook-instance-stopped --notebook-instance-name <notebook-name>
aws sagemaker update-notebook-instance --notebook-instance-name <notebook-name> --lifecycle-config-name autostop-60
aws sagemaker start-notebook-instance --notebook-instance-name <notebook-name>
```

How it was priced. 1 ml.g5.12xlarge at \$7.09 and 2 ml.g5.2xlarge at \$1.52, from 730 hours to 220 and 240, which is \$5,105.50 a month and \$61,266.00 a year.

KIN-1: Kinesis retention from 365 days to 7

Owner: Lior Haddad, Head of Data. **Account:** tp-prod. **Proposed for:** week 3 (April 20 to 24). **Saves:** \$11,040 a year (\$920 a month), rank 27 of 40. **Status:** proposed. **Well-Architected:** one of the four changes that close SUS04, which closes when S3-1 lands; accepted in writing until this engagement.

What we saw. No consumer reads past seven days (GetRecords by iterator age).

The change. Lower the stream's retention from 365 days to 7 with one call. Firehose already delivers the raw stream to S3, which is where "ML replay" can read it, and the iterator age on GetRecords shows that no consumer reads anything older than seven days. The retention charge for data older than seven days stops with the change. **Rollback:** raise the retention again. Records older than seven days are not recovered by that, which is why the S3 copy is checked first (a count of records for one hour on each side).

BASH

```
aws kinesis decrease-stream-retention-period --stream-name <stream-name> --retention-period-hours 168
```

How it was priced. 40,000 GB at \$0.023 a GB-month, which is \$920.00 a month and \$11,040.00 a year; the unit price is from the AWS Price List bulk file.

S3-4: Old exports and CI artifacts expired

Owner: Kofi Brandvold, Head of Platform. **Account:** tp-prod. **Proposed for:** week 3 (April 20 to 24). **Saves:** \$4,129 a year (\$344 a month), rank 38 of 40. **Status:** proposed.

What we saw. Last-accessed dates: none read since 2025.

The change. Expire the old exports and CI artifacts, 16 TB that nothing has read since 2025. The data team lists the prefixes from an S3 Inventory report, each prefix's owner confirms it, and the CloudTrail data events on the export buckets are checked for a read before the rule is saved. Then a lifecycle expiration rule goes on each confirmed prefix, merged into the bucket's existing rules (the API call replaces a bucket's whole lifecycle configuration). **Rollback:** none for an expired object, which is why the prefix list is signed off first.

How it was priced. 16 TB at \$0.021, which is \$344.06 a month and \$4,128.77 a year.

NET-2: Public IPv4 addresses from 20 to 6

Owner: Kofi Brandvold, Head of Platform. **Account:** tp-prod. **Proposed for:** week 3 (April 20 to 24). **Saves:** \$613 a year (\$51 a month), rank 40 of 40. **Status:** proposed.

What we saw. Elastic IPs on stopped instances and the old NAT.

The change. Release 14 of the 20 public IPv4 addresses: the Elastic IPs left on stopped instances and on the old NAT gateway. Each is checked against tidepost-cdk, the route tables, and the partners' allow-lists first. **Rollback:** none for the address itself, since a released Elastic IP cannot be taken back and a new one is a different address. That is the reason to check the allow-lists.

How it was priced. 14 addresses × \$0.005 × 730 hours, which is \$51.10 a month and \$613.20 a year.

EBS-2: Unattached volumes archived, then deleted

Owner: Kofi Brandvold, Head of Platform. **Account:** tp-prod. **Proposed for:** week 3 (April 20 to 24). **Saves:** \$9,677 a year (\$806 a month), rank 28 of 40. **Status:** proposed.

What we saw. Detached for more than 30 days.

The change. Snapshot each volume that has been detached for more than 30 days, move the snapshot to the archive tier, and only then delete the volume. The 9 TB are priced at the archive tier's full size, since an archived snapshot is billed as a full copy. **Rollback:** restore a volume from the archived snapshot, which takes up to a few days in the archive tier.

BASH

```
# From the owner-signed list, per volume.
SNAP=$(aws ec2 create-snapshot --volume-id <volume-id> --description "EBS-2 unattached volume"
--query SnapshotId --output text)
aws ec2 wait snapshot-completed --snapshot-ids "$SNAP"
aws ec2 modify-snapshot-tier --snapshot-id "$SNAP" --storage-tier archive
aws ec2 delete-volume --volume-id <volume-id>
```

How it was priced. 9 TB from \$0.10 to \$0.0125 (archive tier), which is \$806.40 a month and \$9,676.80 a year.

EBS-3: Old snapshot chains deleted

Owner: Kofi Brandvold, Head of Platform. **Account:** tp-prod. **Proposed for:** week 5 (May 4 to 8). **Saves:** \$31,949 a year (\$2,662 a month), rank 9 of 40. **Status:** proposed.

What we saw. The snapshot chains' unique blocks, from the EBS direct APIs.

The change. Stop the cron job that creates snapshots and never deletes them, then delete its snapshot chains past 35 days, oldest first. AWS Backup in tp-backup governs from here, so before any chain is deleted, each volume is checked for a recovery point in AWS Backup. **The honest limit:** EBS snapshots are incremental, so deleting one frees only the blocks no other snapshot references. The estimate counts the unique blocks, read from the EBS direct APIs, not the billed size. **Rollback:** none for a deleted snapshot, so the AWS Backup check comes first.

How it was priced. 52 TB of unique blocks at \$0.05, which is \$2,662.40 a month and \$31,948.80 a year.

ECR-1: ECR lifecycle policy

Owner: Kofi Brandvold, Head of Platform. **Account:** tp-prod. **Proposed for:** week 3 (April 20 to 24). **Saves:** \$5,284 a year (\$440 a month), rank 36 of 40. **Status:** proposed.

What we saw. Images never pulled in 90 days.

The change. Add a lifecycle policy to each repository that expires images no one has pulled in 90 days, with a floor that always keeps the images the running workloads use. The owners export the list of images in use from the clusters first. **Rollback:** an expired image is rebuilt from source by the pipeline.

How it was priced. 4.3 TB at \$0.10, which is \$440.32 a month and \$5,283.84 a year.

The register

The waste register at estimate.

ID	Change	Account	Owner	Per month	Per year
SM-2	Notebook auto-stop	tp-sandbox-ml	Yuki Sorensen	\$5,106	\$61,266
KIN-1	Kinesis retention from 365 days to 7	tp-prod	Lior Haddad	\$920	\$11,040
S3-4	Old exports and CI artifacts expired	tp-prod	Kofi Brandvold	\$344	\$4,129
NET-2	Public IPv4 addresses from 20 to 6	tp-prod	Kofi Brandvold	\$51	\$613
EBS-2	Unattached volumes archived, then deleted	tp-prod	Kofi Brandvold	\$806	\$9,677
EBS-3	Old snapshot chains deleted	tp-prod	Kofi Brandvold	\$2,662	\$31,949
ECR-1	ECR lifecycle policy	tp-prod	Kofi Brandvold	\$440	\$5,284
Total				\$10,330	\$123,958

Compute and machine-learning sizing

\$270,134 a year (10 changes) is compute that is larger than its work or always on when its work is not. This is the largest section. The embedding endpoint alone is \$97,674 a year: it runs on four GPUs and uses one. The staging environment is the second theme. It was built to be half of production and runs a load test at a fraction of that, and it sits in the founders' account until the rebuild in tp-staging.

The embedding endpoint and the classifier

SM-1a: Embedding endpoint to ml.g5.2xlarge

Owner: Yuki Sorensen, Head of Machine Learning. **Account:** tp-prod. **Proposed for:** week 4 (April 27 to May 1). **Saves:** \$97,674 a year (\$8,140 a month), rank 2 of 40. **Status:** proposed. **Well-Architected:** closes COST06, the last of the three changes it needs; closes SUS02, the last of the two changes it needs; accepted in writing until this engagement.

What we saw. GPU utilization by device: device 0 at 31% on average, devices 1 to 3 idle.

The change. Move the embedding endpoint from ml.g5.12xlarge to ml.g5.2xlarge at the same instance count. GPU utilization by device shows one of four GPUs doing the work, device 0 at 31% on average, with the other three idle, so the smaller instance has the one GPU the model uses. Create a new endpoint configuration and update the endpoint, which SageMaker performs as a blue/green deployment; run the embedding load test against the new configuration before the update. **Rollback:** update the endpoint back to the retained configuration, which is a second blue/green deployment.

How it was priced. 2 instances × (\$7.090 – \$1.515) × 730 hours, which is \$8,139.50 a month and \$97,674.00 a year; the unit price is from the AWS Price List bulk file.

SM-1b: Embedding endpoint scales on its backlog

Owner: Yuki Sorensen, Head of Machine Learning. **Account:** tp-prod. **Proposed for:** week 5 (May 4 to 8). **Saves:** \$6,636 a year (\$553 a month), rank 33 of 40. **Status:** proposed.

What we saw. The asynchronous queue's backlog: empty 70% of the day; the floor is one, not zero: one instance stays warm because a cold start (an instance launch and the model load) takes about six minutes, and new posts must be embedded within two minutes to reach the ranking feed.

The change. Scale the asynchronous endpoint on its backlog, between one and four instances: target tracking on `ApproximateBacklogSizePerInstance`, with an alarm on the age of the oldest queued request, because a scale-up from one instance takes minutes. The backlog is empty 70% of the day, so the endpoint rests at its minimum through the quiet hours. This change is priced after SM-1a, on the smaller instance. **Rollback:** set the minimum back to the current count.

How it was priced. $(2 - 1.5) \text{ instances} \times \$1.515 \times 730 \text{ hours}$, which is \$552.97 a month and \$6,635.70 a year; the unit price is from the AWS Price List bulk file.

SM-3: Moderation classifier to Inferentia2

Owner: Yuki Sorensen, Head of Machine Learning. **Account:** tp-prod. **Proposed for:** when its condition is met; the owner sets the date at the readout. **Saves:** \$7,323 a year (\$610 a month), rank 32 of 40. **Status:** proposed.

What we saw. The classifier is a ViT-B/16 that compiles for Neuron.

The change. Compile the moderation classifier, a ViT-B/16, for Neuron with `torch-neuronx` and move it to `ml.inf2.xlarge`. It cannot start until two things are done, and both need the machine-learning team's time: the Neuron compile, and a precision check against the labeled March set that Trust and Safety signs off. Traffic then moves gradually through a second production variant. **Rollback:** shift the variant weight back to the `ml.g5.xlarge` variant, which stays up until the check passes.

How it was priced. $2 \text{ instances} \times (\$1.408 - \$0.990) \times 730 \text{ hours}$, which is \$610.28 a month and \$7,323.36 a year.

Staging

EC2-1a: Staging rebuilt at 30% of its size

Owner: Kofi Brandvold, Head of Platform. **Account:** tp-staging. **Proposed for:** when its condition is met; the owner sets the date at the readout. **Saves:** \$55,468 a year (\$4,622 a month), rank 5 of 40. **Status:** proposed.

What we saw. Staging's load tests use 22% of its capacity at peak.

The change. Rebuild staging in tp-staging, the vended account that is empty today, keeping 30% of the old compute, and retire the old staging in tp-prod. The rebuild is written in `tidepost-cdk`, using the CDK work from the Remediation, with a standard-support EKS control plane and a `db.r6g.large` on PostgreSQL 16 in place of the old staging database, which is on PostgreSQL 13 and bills Extended Support on its 8 vCPUs. The old staging's EKS control plane is on Kubernetes 1.30, in extended support, which ends on July 23, 2026; the rebuild retires it before then. It needs

two platform engineers for the rebuild and the cutover, which is why the Analysis does not give it a week. **Rollback:** the old staging stays until the new one has passed the same load tests, and the CDK stack for the old one is deleted only after that.

How it was priced. The old staging (3 c6i.4xlarge, 5 m5.4xlarge, an EKS control plane in extended support, a db.r6g.2xlarge with PostgreSQL 13 Extended Support on 8 vCPUs) against 30% of its compute, a standard-support control plane, and a db.r6g.large, which is \$4,622.34 a month and \$55,468.08 a year.

EC2-1b: Staging on weekdays only

Owner: Kofi Brandvold, Head of Platform. **Account:** tp-staging. **Proposed for:** when its condition is met; the owner sets the date at the readout. **Saves:** \$11,218 a year (\$935 a month), rank 25 of 40. **Status:** proposed.

What we saw. Deploy logs: no staging deploys at night or on weekends since October.

The change. Run the rebuilt staging on weekdays only, 7 AM to 7 PM Central: scale its node group to zero, and stop its database, outside those hours, with an EventBridge schedule written in CDK. The deploy logs show no staging deploys at night or on weekends since October. A database stopped by schedule restarts by itself after seven days, which a weekend never reaches. It follows EC2-1a, since there is nothing to schedule until the rebuilt staging exists. **Rollback:** disable the schedule.

How it was priced. The rebuilt staging's compute and database, from 730 hours to 260, which is \$934.83 a month and \$11,217.96 a year.

The API tier, the transcoders, and EKS

EC2-2: Developer instances off outside working hours

Owner: Kofi Brandvold, Head of Platform. **Account:** tp-dev. **Proposed for:** week 3 (April 20 to 24). **Saves:** \$27,095 a year (\$2,258 a month), rank 15 of 40. **Status:** proposed. **Well-Architected:** one of the three changes that close COST06, which closes when SM-1a lands; accepted in writing until this engagement.

What we saw. CPU by hour: under 2% outside 8 AM to 6 PM on 11 of 12.

The change. Stop the developer instances outside working hours with the Instance Scheduler on AWS in tp-dev: a schedule from 8 AM to 6 PM Central, the hours the CPU data supports. An instance tagged schedule-exempt stays on, and the platform team reviews the exempt list monthly. **Rollback:** remove the schedule tag from an instance.

How it was priced. 12 m5.2xlarge at \$0.384, from 730 hours to 240, which is \$2,257.92 a month and \$27,095.04 a year.

EC2-3: API tier minimum from 9 to 6

Owner: Marcus Oyelaran, Backend Lead. **Account:** tp-prod. **Proposed for:** week 3 (April 20 to 24). **Saves:** \$11,914 a year (\$993 a month), rank 24 of 40. **Status:** proposed.

What we saw. Request rate and CPU by hour: peaks need 8 instances, nights need 4.

The change. Lower the API tier's Auto Scaling minimum from 9 to 6, two instances in each of the three zones, and add target tracking on request count per target. The request rate and CPU by hour show that peaks need 8 instances and nights need 4, so a minimum of 6 holds a zone-failure reserve through the night without carrying nine. The maximum does not change. The change ships through the target-group swap with automatic rollback. **Rollback:** the swap reverts, or the minimum goes back to nine.

How it was priced. From 9 to 7 c6i.4xlarge on average at \$0.680, which is \$992.80 a month and \$11,913.60 a year.

EC2-4: API tier on Graviton

Owner: Marcus Oyelaran, Backend Lead. **Account:** tp-prod. **Proposed for:** week 4 (April 27 to May 1). **Saves:** \$6,132 a year (\$511 a month), rank 34 of 40. **Status:** proposed.

What we saw. The API is Go; its arm64 build passed the load test.

The change. Move the API tier to Graviton c7g.4xlarge. The API is written in Go, and its arm64 build passed the load test. A new launch template with the arm64 image goes in through the target-group swap, so the c6i.4xlarge instances keep serving until the new ones are healthy. This change is priced after EC2-3, on the average number of instances that EC2-3 leaves. **Rollback:** swap the target group back to the c6i.4xlarge launch template.

How it was priced. 7 instances $\times$ (\$0.680 – \$0.580) $\times$ 730 hours, which is \$511.00 a month and \$6,132.00 a year.

EC2-5: Transcoders scale on queue depth

Owner: Kofi Brandvold, Head of Platform. **Account:** tp-prod. **Proposed for:** week 4 (April 27 to May 1). **Saves:** \$19,763 a year (\$1,647 a month), rank 21 of 40. **Status:** proposed.

What we saw. Queue depth: zero for 61% of the day.

The change. Scale the transcoder fleet on queue depth, with an Auto Scaling group on the transcode queue's backlog per instance and a minimum of two instances. The fleet is g4dn.2xlarge, and the queue is empty for 61% of the day. **Rollback:** set the minimum back to the current fixed count.

How it was priced. From 6 to 3 g4dn.2xlarge on average at \$0.752, which is \$1,646.88 a month and \$19,762.56 a year.

EKS-1: Karpenter consolidation, production nodes

Owner: Kofi Brandvold, Head of Platform. **Account:** tp-prod. **Proposed for:** week 4 (April 27 to May 1). **Saves:** \$26,911 a year (\$2,243 a month), rank 16 of 40. **Status:** proposed.

What we saw. Pod requests sum to 58% of node capacity.

The change. Turn on Karpenter consolidation for the production node pool and bring the nodes down from 12 to 8, with disruption budgets and a PodDisruptionBudget on every service that does not have one, so consolidation never evicts more than a service can lose. Pod requests sum to 58% of node capacity, which leaves room for 8 nodes. Pods with hard zone constraints are listed first, since they decide how far consolidation can go. **Rollback:** raise the node pool's limits and minimum, and Karpenter adds nodes back.

YAML

```
# NodePool disruption block, in tidepost-cdk
disruption:
  consolidationPolicy: WhenEmptyOrUnderutilized
  consolidateAfter: 5m
  budgets:
    - nodes: "20%"
```

How it was priced. 4 m5.4xlarge at \$0.768 x 730 hours, which is \$2,242.56 a month and \$26,910.72 a year.

Databases, caches, and the telemetry stream

\$117,778 a year (7 changes) is capacity provisioned above what the stores use. The evidence is the same in each case: a peak that sits well below the instance, shard, or table capacity behind it. These are the changes where an owner's caution is most reasonable, because a resize touches a store that serves users. Each card says what the rollback costs.

Telemetry and online features

KIN-2: Kinesis from 100 shards to 20

Owner: Lior Haddad, Head of Data. **Account:** tp-prod. **Proposed for:** week 4 (April 27 to May 1). **Saves:** \$25,795 a year (\$2,150 a month), rank 18 of 40. **Status:** proposed.

What we saw. Peak ingest is 3.1 MB a second; each shard takes 1 MB.

The change. Take the stream from 100 shards to 20 and aggregate records in the collector, so each put carries up to 25 KB instead of 1 KB. Peak ingest is 3.1 MB a second and a shard takes 1 MB, so 20 shards carry the peak with room to spare. UpdateShardCount will not cut a stream by more than half in one call, so the change runs in steps, with the iterator-age alarm watched after each.

Rollback: raise the shard count again, which is also a staged operation and takes the same time.

How it was priced. 80 shards $\times$ (\$0.015 + \$0.02) $\times$ 730 hours, and PUT units $\div$ 25, which is \$2,149.56 a month and \$25,794.69 a year.

DDB-1: Provisioned capacity on the two steady tables

Owner: Lior Haddad, Head of Data. **Account:** tp-prod. **Proposed for:** week 4 (April 27 to May 1).

Saves: \$25,058 a year (\$2,088 a month), rank 19 of 40. **Status:** proposed.

What we saw. Request rate by minute: the two feature tables vary 30% across the day.

The change. Switch the two steady tables, which carry 85% of the traffic, from on-demand to provisioned capacity with auto scaling, and leave the bursty tables on-demand. The request rate by minute shows the two feature tables vary 30% across the day, so auto scaling follows them. The target utilization is set so provisioned capacity runs at about 140% of the mean, as priced. DynamoDB allows one capacity-mode change per table every 24 hours. **Rollback:** switch the table back to on-demand, once the 24 hours have passed.

How it was priced. 85% of \$4,125.00 on demand, against provisioned units at 140% of the mean, which is \$2,088.17 a month and \$25,058.00 a year.

Databases and caches

RDS-1: Read replicas down one size

Owner: Dev Raman, Co-founder and Chief Technology Officer. **Account:** tp-prod. **Proposed for:** week 4 (April 27 to May 1). **Saves:** \$7,866 a year (\$656 a month), rank 30 of 40. **Status:** proposed.

What we saw. Buffer cache hit 99.6% with a 21 GB working set; CPU peaks at 31%.

The change. Modify the two read replicas from db.r6g.2xlarge to db.r6g.xlarge, one at a time, with the application reading from the other. The working set is 21 GB against the memory of the smaller class, the buffer cache hit rate is 99.6%, and CPU peaks at 31%. **Rollback:** modify the replica back; it is a restart of that replica only.

How it was priced. 2 $\times$ (\$0.899 – \$0.450) $\times$ 730 hours, which is \$655.54 a month and \$7,866.48 a year.

RDS-2: Writer down one size

Owner: Dev Raman, Co-founder and Chief Technology Officer. **Account:** tp-prod. **Proposed for:** week 5 (May 4 to 8). **Saves:** \$15,759 a year (\$1,313 a month), rank 23 of 40. **Status:** proposed.

What we saw. CPU peaks at 38%; memory pressure none.

The change. Modify the Multi-AZ writer from db.r6g.4xlarge to db.r6g.2xlarge. CPU peaks at 38% and there is no memory pressure. RDS resizes the standby first and then fails over, so the change costs one failover of about a minute or two; it is made in a change window, never inside a busy hour. Tidepost plans for growth after the round, and a smaller writer has less headroom for it, so this is the one change in the section whose price is a decision about growth and not a measurement. **Rollback:** modify the writer back, which is a second failover.

How it was priced. $(\$3.597 - \$1.798) \times 730$ hours, which is \$1,313.27 a month and \$15,759.24 a year.

CACHE-1: ElastiCache nodes down one size

Owner: Kofi Brandvold, Head of Platform. **Account:** tp-prod. **Proposed for:** week 4 (April 27 to May 1). **Saves:** \$28,768 a year (\$2,397 a month), rank 13 of 40. **Status:** proposed.

What we saw. BytesUsedForCache peaks at 29 GB a shard.

The change. Modify the cache nodes from cache.r6g.4xlarge to cache.r6g.2xlarge, one shard at a time, with ElastiCache's online scaling. BytesUsedForCache peaks at 29 GB a shard, against 105 GB of memory on the current node, so a node half the size still holds the peak with room to spare.

Rollback: scale the shard back up online.

How it was priced. $4 \text{ nodes} \times (\$1.642 - \$0.821) \times 730$ hours, which is \$2,397.32 a month and \$28,767.84 a year.

CACHE-2: Redis OSS to Valkey

Owner: Kofi Brandvold, Head of Platform. **Account:** tp-prod. **Proposed for:** when its condition is met; the owner sets the date at the readout. **Saves:** \$5,754 a year (\$479 a month), rank 35 of 40. **Status:** proposed.

What we saw. Valkey 8 keeps the Redis OSS 7 API.

The change. Change the engine from Redis OSS 7.1 to Valkey 8 on the same nodes. Valkey 8 keeps the Redis OSS 7 API, and ElastiCache upgrades a replication group in place, which costs a failover per shard. The client library is tested against Valkey in staging first, because that is where an incompatibility would show; until staging is rebuilt, the test runs against a throwaway cluster.

This change is priced after CACHE-1, on the smaller nodes. **Rollback:** Valkey cannot be downgraded back to Redis OSS in place; the rollback is a restore from the snapshot taken before the change.

How it was priced. 4 nodes $\times$ (\$0.821 – \$0.6568) $\times$ 730 hours, which is \$479.46 a month and \$5,753.57 a year.

OS-1: OpenSearch data nodes down one size

Owner: Lior Haddad, Head of Data. **Account:** tp-prod. **Proposed for:** week 5 (May 4 to 8). **Saves:** \$8,778 a year (\$731 a month), rank 29 of 40. **Status:** proposed.

What we saw. JVM memory pressure peaks at 41%.

The change. Change the three data nodes from r6g.2xlarge.search to r6g.xlarge.search; OpenSearch makes the change with a blue/green deployment, so the domain stays up. JVM memory pressure peaks at 41%. The search team must be comfortable with the change before it is made, because it is their index. **Rollback:** change the instance type back, which is another blue/green deployment.

How it was priced. 3 nodes $\times$ (\$0.669 – \$0.335) $\times$ 730 hours, which is \$731.46 a month and \$8,777.52 a year.

Moderation and language

\$125,322 a year (3 changes) is the cost of looking at content twice. Every video already passes the frame path. Since July 2025 it also passes Rekognition Video, which was added after a graphic video reached timelines, and every video is transcribed whether or not anyone speaks in it. Moderation is the work that keeps Tidepost's timelines safe, so the Analysis changes how much looks at what, never whether something does. Amara Delacroix owns all three, and none is made without Trust and Safety's yes.

The three changes

MOD-1: Rekognition Video for new accounts and reported videos only

Owner: Amara Delacroix, Head of Trust and Safety. **Account:** tp-prod. **Proposed for:** week 4 (April 27 to May 1). **Saves:** \$83,160 a year (\$6,930 a month), rank 3 of 40. **Status:** proposed.

What we saw. The moderation log for March: every video already passes the frame path, and the second path flagged 0.02% of older accounts' videos the frame path missed.

The change. Send a video to Rekognition Video only when its account is under 90 days old or it has been reported; the frame path covers the rest. In the moderation pipeline's Lambda chain, add one branch ahead of the second path: `account_age_days < 90 or reported`. The moderation log for March shows that every video already passes the frame path, and that the second path flagged 0.02% of older accounts' videos which the frame path missed. That 0.02% is the coverage the change gives up, and Trust and Safety should decide whether it is acceptable. The branch ships behind a flag. **Rollback:** switch the flag, and the second path takes every video again; videos skipped in the meantime are stored, so Rekognition Video can run over them afterward.

How it was priced. 173,250 minutes a month at \$0.10, times the 40% of videos from older accounts that nobody reports, which is \$6,930.00 a month and \$83,160.00 a year; the unit price is from the AWS Price List bulk file.

MOD-2: Frame sampling every two seconds

Owner: Amara Delacroix, Head of Trust and Safety. **Account:** tp-prod. **Proposed for:** week 5 (May 4 to 8). **Saves:** \$37,422 a year (\$3,119 a month), rank 7 of 40. **Status:** proposed.

What we saw. Frame sampler configuration; Rekognition image-moderation usage by day.

The change. Sample frames every two seconds instead of every second, in the frame sampler's configuration. This change is priced after MOD-1: once MOD-1 sends fewer videos to the second path, the frame path carries most videos alone, and halving it halves what that path sees. It is the one change in this report that trades coverage for cost directly, so we list it second and make no recommendation beyond the price. **Rollback:** set the interval back to one second.

How it was priced. 11.58 million images a month to 6.39 million, at the published Group 2 tiers, which is \$3,118.50 a month and \$37,422.00 a year; the unit price is from the AWS Price List bulk file.

TR-1: Transcribe only videos with speech

Owner: Amara Delacroix, Head of Trust and Safety. **Account:** tp-prod. **Proposed for:** week 3 (April 20 to 24). **Saves:** \$4,740 a year (\$395 a month), rank 37 of 40. **Status:** proposed.

What we saw. A week of Transcribe output: 38% of transcripts were empty.

The change. Send a video to Transcribe only when its audio track has detected speech: a voice-activity check in the transcode step, ahead of the Transcribe job. A week of Transcribe output shows that 38% of transcripts were empty, so those jobs produce nothing. **Rollback:** let every video through the check, and every video is transcribed again.

How it was priced. 173,250 minutes at \$0.006 a minute, less the 38% with no speech, which is \$395.01 a month and \$4,740.12 a year; the unit price is from the AWS Price List bulk file.

Storage and the data lake

\$206,898 a year (5 changes) is data kept in a storage class priced for a use it does not have. S3 Standard holds 1,087 TB (709 originals, 207 renditions, 151 raw telemetry, 20 exports and artifacts), and nothing is in any other class. Each S3 change is priced on the Standard tier it really leaves: Standard bills \$0.021 a GB-month above its first 500 TB, so the first 587 TB to leave come off that top tier. S3-4, S3-3, and S3-1 together take 325 TB of it; S3-2, priced last, takes the remaining 262 TB, and its last 301 TB come off the \$0.022 tier. The changes differ in how fast the data comes back, and each card says how fast. EBS-1 sits here too: it is storage, and it changes nothing about how the volumes behave.

Telemetry

S3-1: Telemetry to Parquet, raw JSON to Deep Archive

Owner: Lior Haddad, Head of Data. **Account:** tp-prod. **Proposed for:** week 5 (May 4 to 8). **Saves:** \$30,640 a year (\$2,553 a month), rank 12 of 40. **Status:** proposed. **Well-Architected:** closes PERF03, the last of the two changes it needs; closes SUS04, the last of the four changes it needs; accepted in writing until this engagement.

What we saw. Athena query history: every query reads one to seven days.

The change. Convert telemetry to Parquet at ingest, convert the 151 TB of raw JSON already in S3, and move the raw JSON to S3 Glacier Deep Archive. Firehose record format conversion writes new data as Parquet; a Glue job converts the existing partitions, partition by partition, and each partition's row count is checked against its JSON before the JSON moves. The Athena query history shows every query reads one to seven days, so nothing needs the old JSON live. Deep Archive restores in hours, not milliseconds, and Athena cannot read an object in Deep Archive, which is why ATH-1 comes after this change. **Rollback:** restore a partition from Deep Archive, which takes hours and has a retrieval charge.

How it was priced. 151 TB at \$0.021 to 18.9 TB of Parquet, the JSON at \$0.00099, plus Firehose format conversion, which is \$2,553.32 a month and \$30,639.78 a year. One-time charge: \$461.

ATH-1: Athena on partitioned Parquet

Owner: Lior Haddad, Head of Data. **Account:** tp-prod. **Proposed for:** week 5 (May 4 to 8). **Saves:** \$17,340 a year (\$1,445 a month), rank 22 of 40. **Status:** proposed. **Well-Architected:** one of the two changes that close PERF03, which closes when S3-1 lands; accepted in writing until this engagement.

What we saw. Each query scans whole days of JSON for a few columns.

The change. Point Athena at the partitioned Parquet table, with partition projection on the date and hour, and give the data team's workgroup a limit on the bytes one query may scan. Each query today scans whole days of JSON for a few columns; Parquet reads only the columns asked for, and partitions limit it to the days asked for. The saving is priced after S3-1, on the volume that Parquet leaves. **Rollback:** the JSON table stays in the catalog, but its objects are in Deep Archive after S3-1, so the rollback is the restore in S3-1.

How it was priced. From 330 TB to 41 TB scanned a month at \$5.00 a TB, which is \$1,445.00 a month and \$17,340.00 a year.

The media buckets

S3-2: Originals over 90 days to Glacier Instant Retrieval

Owner: Amara Delacroix, Head of Trust and Safety. **Account:** tp-prod. **Proposed for:** when its condition is met; the owner sets the date at the readout. **Saves:** \$120,787 a year (\$10,066 a month), rank 1 of 40. **Status:** proposed.

What we saw. S3 Storage Lens: originals older than 90 days are read 0.3% a month.

The change. Transition originals older than 90 days to S3 Glacier Instant Retrieval with a lifecycle rule on the originals bucket. Instant Retrieval returns an object in milliseconds, so a takedown or legal request stays fast; S3 Storage Lens shows originals older than 90 days are read 0.3% a month. The rule is written once, with the retention schedule counsel signs, so the bucket has one lifecycle owner and one change. It cannot be written before counsel signs. **Rollback:** a transition cannot be reversed by the rule, but an object can be copied back into Standard; the price assumes 1.5 TB a month is read back at \$0.03 a GB.

How it was priced. 563 TB from Standard (the first 262 TB at \$0.021, the last 301 at \$0.022, after the in-place changes) to \$0.004, plus 1.5 TB a month read back at \$0.03, which is \$10,065.55 a month and \$120,786.55 a year. One-time charge: \$192.

S3-3: Renditions over 60 days to Glacier Instant Retrieval

Owner: Amara Delacroix, Head of Trust and Safety. **Account:** tp-prod. **Proposed for:** week 5 (May 4 to 8). **Saves:** \$30,734 a year (\$2,561 a month), rank 10 of 40. **Status:** proposed. **Well-Architected:** one of the four changes that close SUS04, which closes when S3-1 lands; accepted in writing until this engagement.

What we saw. S3 Inventory of April 6 by size and age; CloudFront logs: renditions older than 60 days serve 2% of views and 10 TB a month is read back.

The change. Transition renditions older than 60 days, 128 KB and up, to S3 Glacier Instant Retrieval, which serves CloudFront’s fetch of an old segment in milliseconds. The S3 Inventory of April 6 gives the input: 128,000,000 objects and 176.7 TB at 128 KB and up, 124.0 TB of them older than 150 days; the 22,400,000 smaller objects (1.3 TB) stay in Standard, as a lifecycle rule leaves them by default. CloudFront logs show renditions older than 60 days serve 2% of views and 10 TB a month is read back. Priced on the same inventory with all three of its tiers, Intelligent-Tiering saves \$23,253 a year: it reaches the same prices at the same ages a lifecycle rule would, but monitors 128,000,000 objects at \$320 a month and holds the youngest band at the Standard price for its first 30 days. Standard-IA saves \$15,981. This change saves the most, and it stops beating Standard-IA only above 70 TB a month read back. The one-time transition requests are the \$2,560 listed under [One-time charges](#). **Rollback:** remove the rule; objects already moved stay in Glacier Instant Retrieval until copied back, and an object moved out within 90 days bills the rest of its 90-day minimum.

How it was priced. 176.7 TB of 128 KB and up from \$0.021 to \$0.004, plus 10 TB a month read back at \$0.03 and the transition requests for each month’s 10,400,000 objects that turn 60 days old, which is \$2,561.14 a month and \$30,733.70 a year. One-time charge: \$2,560.

Volumes

EBS-1: gp2 to gp3

Owner: Kofi Brandvold, Head of Platform. **Account:** tp-prod. **Proposed for:** week 3 (April 20 to 24). **Saves:** \$7,397 a year (\$616 a month), rank 31 of 40. **Status:** proposed.

What we saw. CloudWatch over 30 days: no volume peaked above 3,000 IOPS or 125 MiB/s, so gp3’s included baseline covers every one with nothing provisioned.

The change. Convert the 30.1 TB of gp2 volumes to gp3 in place, with `modify-volume`. No volume peaked above 3,000 IOPS or 125 MiB/s over 30 days, and gp3’s included baseline is 3,000 IOPS and 125 MiB/s, so nothing is provisioned beyond the baseline. gp3 allows up to 80,000 IOPS and 2,000 MiB/s, far above anything these volumes use. The conversion runs with the volume attached and in use. **Rollback:** modify the volume back to gp2 after the six-hour wait AWS requires between modifications.

BASH

```
aws ec2 modify-volume --volume-id <volume-id> --volume-type gp3
```

How it was priced. 30.1 TB × (\$0.10 – \$0.08), which is \$616.45 a month and \$7,397.38 a year.

Logging and metrics

\$90,517 a year (3 changes) is CloudWatch charging for logs and metrics that no one reads. The API logs at DEBUG, log groups never expire, and the custom metrics carry a per-post dimension that no dashboard or alarm uses. Audit logs are a different thing and are not touched: the organization trail and Config history live 400 days in tp-log-archive, a retention the security engagement set, so only application logs change here.

The three changes

LOG-1: API log level to INFO

Owner: Marcus Oyelaran, Backend Lead. **Account:** tp-prod. **Proposed for:** week 3 (April 20 to 24). **Saves:** \$30,720 a year (\$2,560 a month), rank 11 of 40. **Status:** proposed.

What we saw. Log groups by level: 76% of bytes are DEBUG.

The change. Change the API's log level from DEBUG to INFO in production, in the service's configuration in tidepost-cdk. The log groups by level show 76% of bytes are DEBUG. A pull request lowers the default, and a flag raises one instance to DEBUG for an hour when an engineer is chasing a problem, so debugging is still possible without paying for it all day. **Rollback:** set the default back to DEBUG.

How it was priced. From 6.6 TB to 1.6 TB a month at \$0.50 a GB, which is \$2,560.00 a month and \$30,720.00 a year.

LOG-2: Application logs kept 30 days

Owner: Sasha Ilyin, Head of Security and Compliance. **Account:** tp-prod. **Proposed for:** week 5 (May 4 to 8). **Saves:** \$32,197 a year (\$2,683 a month), rank 8 of 40. **Status:** proposed. **Well-Architected:** one of the four changes that close SUS04, which closes when S3-I lands; accepted in writing until this engagement.

What we saw. Audit logs live 400 days in tp-log-archive (the security engagement), so only application logs change.

The change. Keep application log groups 30 days. Counsel needs January to March preserved, so the export comes first: CloudWatch Logs export tasks write those months to the export bucket in S3 Standard, the bucket's default class, and a lifecycle rule on the export prefix moves each object to S3 Glacier Deep Archive the next day. Check the object counts against the log groups, and have counsel confirm the copy. Only then set the retention policy. The export's writes, its day

in S3 Standard, and the Deep Archive transition requests are a one-time charge ([one-time charges](#)). This change is priced after LOG-1, on the smaller volume LOG-1 leaves. **Rollback:** none for an expired log event, which is why counsel's copy and the object-count check come first.

BASH

```
aws logs create-export-task --log-group-name <log-group> --from <start-ms> --to <end-ms> --  
destination <export-bucket> --destination-prefix log-export  
aws logs put-retention-policy --log-group-name <log-group> --retention-in-days 30
```

How it was priced. 89.8 TB stored to 1.8 TB at \$0.03, plus 20 TB in Deep Archive at \$0.00099, which is \$2,683.08 a month and \$32,197.02 a year. One-time charge: \$143.

LOG-3: Custom metrics from 12,000 to 3,000

Owner: Marcus Oyelaran, Backend Lead. **Account:** tp-prod. **Proposed for:** week 3 (April 20 to 24). **Saves:** \$27,600 a year (\$2,300 a month), rank 14 of 40. **Status:** proposed.

What we saw. 9,000 metrics carry a post id; no dashboard or alarm reads them.

The change. Stop emitting the per-post dimension on the custom metrics, which takes the count from 12,000 to 3,000. Nine thousand metrics carry a post id, and no dashboard or alarm reads them, which an export of every dashboard and alarm definition confirms before the change. The change is in the metric publisher, in a pull request. CloudWatch bills a metric by the hours it exists, so the saving starts as the old ones stop receiving data. **Rollback:** emit the dimension again; its history starts from that day.

How it was priced. The published metric tiers before and after, which is \$2,300.00 a month and \$27,600.00 a year.

Networking and delivery

\$84,841 a year (3 changes) is spent moving bytes a longer way than they need to go. NAT gateways bill by the gigabyte for traffic to AWS's own services, and CloudFront bills by the gigabyte for video that a phone screen cannot show at full size. Public IPv4 addresses are in the waste register (NET-2).

The three changes

NET-1: S3 and DynamoDB gateway endpoints

Owner: Kofi Brandvold, Head of Platform. **Account:** tp-prod. **Proposed for:** week 3 (April 20 to 24). **Saves:** \$26,763 a year (\$2,230 a month), rank 17 of 40. **Status:** proposed.

What we saw. VPC flow logs: 80% of NAT bytes go to S3 and DynamoDB prefixes.

The change. Add gateway endpoints for S3 and DynamoDB to the production VPC and associate them with the private route tables in all three zones. Gateway endpoints cost nothing. The VPC flow logs show 80% of NAT bytes go to S3 and DynamoDB prefixes. A gateway endpoint carries same-Region traffic only, so a bucket in another Region still goes through NAT, which the flow logs identify before the change. **Rollback:** remove the endpoint; routes fall back to the NAT gateways.

TYPESCRIPT

```
// tidepost-cdk, production VPC
vpc.addGatewayEndpoint('S3', { service: ec2.GatewayVpcEndpointAwsService.S3 });
vpc.addGatewayEndpoint('DynamoDB', { service: ec2.GatewayVpcEndpointAwsService.DYNAMODB });
```

How it was priced. 80% of 60.5 TB processed at \$0.045 a GB, which is \$2,230.27 a month and \$26,763.26 a year.

CDN-1: Rendition ladder: 720p on cellular

Owner: Bea Okafor, Mobile Engineering Lead. **Account:** tp-prod. **Proposed for:** week 4 (April 27 to May 1). **Saves:** \$55,443 a year (\$4,620 a month), rank 6 of 40. **Status:** proposed.

What we saw. Bytes per view by rendition: 1080p on a phone screen is 41% of bytes.

The change. Change the rendition ladder in the app: 720p on cellular, and no 1080p rendition for vertical clips. The bytes-per-view data by rendition shows 1080p on a phone screen is 41% of bytes. The change ships with the next app release, and the saving arrives as viewers update; the price is the saving once every viewer has the new version, which is why week six measures it per view. Bea Okafor owns the release. **Rollback:** a server-side flag returns the old ladder to every app version that checks it.

How it was priced. 470 TB less 24% at the published North America tiers, once every viewer has the new app, which is \$4,620.29 a month and \$55,443.46 a year.

CDN-1b: One rendition fewer to transcode

Owner: Kofi Brandvold, Head of Platform. **Account:** tp-prod. **Proposed for:** week 5 (May 4 to 8). **Saves:** \$2,635 a year (\$220 a month), rank 39 of 40. **Status:** proposed.

What we saw. The 1080p rendition takes 13% of transcoding time.

The change. Stop transcoding the 1080p rendition for vertical clips, the uploads that CDN-1 no longer serves it for. The 1080p rendition takes 13% of transcoding time. The saving is priced after EC2-5, as a share of the transcoder hours that EC2-5 leaves. It follows CDN-1 by design: the pipeline stops producing a rendition only once the app has stopped asking for it. **Rollback:** turn the rendition back on in the transcode configuration.

How it was priced. 0.4 g4dn.2xlarge on average at $\$0.752 \times 730$ hours (priced after EC2-5), which is \$219.58 a month and \$2,635.01 a year.

Commitments

Two Savings Plans, \$34,956 a year at the estimate, are sized last, on what every other change leaves. A commitment is priced on the usage that will still be running after the sizing and scheduling changes, because a plan bought on today's usage and then outgrown by a rightsizing would bill for compute that no longer exists. The estate has no Savings Plan, Reserved Instance, or private pricing today.

The floor. The floor is the always-on usage left after every in-place and waiting change in this report, counting the minimums and not the averages. Compute: \$14.26 an hour On-Demand, or \$10.38 an hour at the one-year Compute Savings Plan rate. SageMaker hosting: \$6.78 an hour On-Demand, or \$5.18 at the one-year SageMaker Savings Plan rate. Each plan commits to a fraction of the plan-rate cost, so that it stays fully used even if a waiting change lands or the floor moves.

What a Savings Plan can be undone by. A Savings Plan of \$100 an hour or less can be returned within seven days of purchase, in the same calendar month. After that it cannot be canceled. A plan bought on the last day of a month therefore has almost no return window, and a plan that is returned must be returned in the month it was bought. Both plans here are far below \$100 an hour, so the rule applies to them, and Gwen Tallis reads the floor again the day before each purchase.

The two plans

SP-1: Compute Savings Plan

Owner: Gwen Tallis, Vice President, Finance. **Account:** tp-management. **Proposed for:** week 4 (April 27 to May 1). **Saves:** \$23,764 a year (\$1,980 a month), rank 20 of 40. **Status:** proposed. **Well-Architected:** closes COST07; accepted in writing until this engagement.

What we saw. The floor after every change on the roadmap is \$14.260 an hour On-Demand; a plan of \$100 an hour or less can be returned within seven days in the same month.

The change. Buy a one-year, No Upfront Compute Savings Plan at \$7.27 an hour in tp-management, the payer, with Savings Plans sharing on so it covers compute in every account. It is sized at 70% of the floor's plan-rate cost. A Compute Savings Plan covers EC2 in any family or Region, including the Graviton API tier from EC2-4 and the EKS nodes Karpenter launches, and also Fargate and Lambda. It is bought after the sizing changes land, so the floor is known. Gwen Tallis buys it; Vantalect never does. **Rollback:** a return within the seven-day, same-month window above, and not after.

How it was priced. \$7.27 an hour covers \$9.9828 of On-Demand usage at the floor’s blended one-year rate (27.2% off), which is \$1,980.33 a month and \$23,763.98 a year.

SP-2: SageMaker Savings Plan

Owner: Gwen Tallis, Vice President, Finance. **Account:** tp-management. **Proposed for:** after SM-1a, on the hosting floor it leaves. **Saves:** \$11,192 a year (\$933 a month), rank 26 of 40. **Status:** proposed.

What we saw. The hosting floor after SM-1a.

The change. Buy a one-year, No Upfront SageMaker Savings Plan at \$4.14 an hour in tp-management, sized at 80% of the hosting floor after SM-1a. SageMaker Savings Plans apply across instance families and sizes, so the plan stays used when SM-3 moves the classifier to ml.inf2.xlarge. It is bought after SM-1a is in place, since SM-1a cuts the largest line it would cover.

Rollback: the same seven-day, same-month return.

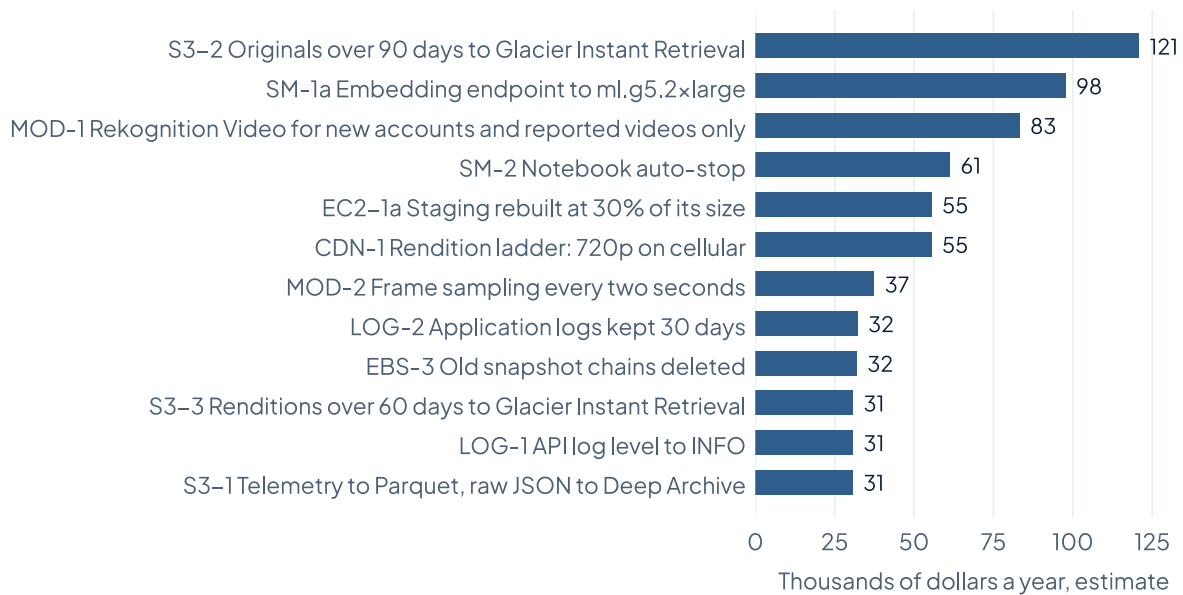
How it was priced. \$4.14 an hour covers \$5.4176 of SageMaker hosting at the floor’s blended rate (23.6% off), which is \$932.65 a month and \$11,191.80 a year.

The commitments, sized on the floor. Each estimate is the On-Demand cost the plan covers, less its hourly fee, times 730 hours and 12 months.

	Compute Savings Plan	SageMaker Savings Plan
The floor, On-Demand, per hour	\$14.26	\$6.78
The floor at the plan rate, per hour	\$10.38	\$5.18
The commitment, per hour	\$7.27	\$4.14
The commitment as a share of the floor at the plan rate	70%	80%
The estimate, per year	\$23,764	\$11,192

The savings roadmap

All 40 changes ranked by what they save a year, at the estimate: \$1,054,404, 50.0% of the April bill. The ten largest are \$606,100. The roadmap is a proposal: the owners decide change by change at tomorrow’s readout, and what they decide, and when, is in the next report.



The twelve largest changes, in thousands of dollars a year at the estimate.

The plan by week

The Analysis proposes the changes in the order that suits the work: the waste and the scheduled changes first, then the sizing, then the data lifecycle and the telemetry, with the commitments after the sizing is in. 5 changes wait on a condition, and each owner sets a date at the readout.

The proposal by week, at estimate. Week 6 (the Cost and Usage Report read and the measurement report) proposes no change.

When	Changes	Count	A year
Week 3: April 20 to 24	TR-1, SM-2, EC2-2, EC2-3, KIN-1, LOG-1, LOG-3, S3-4, NET-1, NET-2, EBS-1, EBS-2, ECR-1	13	\$228,238
Week 4: April 27 to May 1	MOD-1, SM-1a, EC2-4, EC2-5, EKS-1, DDB-1, KIN-2, CDN-1, RDS-1, CACHE-1, SP-1	11	\$400,334
Week 5: May 4 to 8	MOD-2, SM-1b, LOG-2, S3-1, ATH-1, S3-3, CDN-1b, EBS-3, RDS-2, OS-1	10	\$214,090
Waits on a condition: Each owner sets the date at the readout	SM-3, EC2-1a, EC2-1b, S3-2, CACHE-2	5	\$200,550
After SM-1a: On the hosting floor SM-1a leaves	SP-2	1	\$11,192
All		40	\$1,054,404

The ranked register

All 40 changes, ranked. Each year figure is rounded half up and the total is the sum of the rows as printed.

Rank	ID	Change	Owner	Account	Proposed for	A month	A year	Cumulative share
1	S3-2	Originals over 90 days to Glacier Instant Retrieval	Amara Delacroix	tp-prod	when its condition is met; the owner sets the date at the readout	\$10,066	\$120,787	11.5%
2	SM-1a	Embedding endpoint to ml.g5.2xlarge	Yuki Sorensen	tp-prod	week 4 (April 27 to May 1)	\$8,140	\$97,674	20.7%
3	MOD-1	Rekognition Video for new accounts and reported videos only	Amara Delacroix	tp-prod	week 4 (April 27 to May 1)	\$6,930	\$83,160	28.6%
4	SM-2	Notebook auto-stop	Yuki Sorensen	tp-sandbox-ml	week 3 (April 20 to 24)	\$5,106	\$61,266	34.4%
5	EC2-1a	Staging rebuilt at 30% of its size	Kofi Brandvold	tp-staging	when its condition is met; the owner sets the date at the readout	\$4,622	\$55,468	39.7%
6	CDN-1	Rendition ladder: 720p on cellular	Bea Okafor	tp-prod	week 4 (April 27 to May 1)	\$4,620	\$55,443	44.9%
7	MOD-2	Frame sampling every two seconds	Amara Delacroix	tp-prod	week 5 (May 4 to 8)	\$3,119	\$37,422	48.5%
8	LOG-2	Application logs kept 30 days	Sasha Ilyin	tp-prod	week 5 (May 4 to 8)	\$2,683	\$32,197	51.5%
9	EBS-3	Old snapshot chains deleted	Kofi Brandvold	tp-prod	week 5 (May 4 to 8)	\$2,662	\$31,949	54.6%
10	S3-3	Renditions over 60 days to Glacier Instant Retrieval	Amara Delacroix	tp-prod	week 5 (May 4 to 8)	\$2,561	\$30,734	57.5%
11	LOG-1	API log level to INFO	Marcus Oyelaran	tp-prod	week 3 (April 20 to 24)	\$2,560	\$30,720	60.4%
12	S3-1	Telemetry to Parquet, raw JSON to Deep Archive	Lior Haddad	tp-prod	week 5 (May 4 to 8)	\$2,553	\$30,640	63.3%

Rank	ID	Change	Owner	Account	Proposed for	A month	A year	Cumulative share
13	CACHE-1	ElastiCache nodes down one size	Kofi Brandvold	tp-prod	week 4 (April 27 to May 1)	\$2,397	\$28,768	66.0%
14	LOG-3	Custom metrics from 12,000 to 3,000	Marcus Oyelaran	tp-prod	week 3 (April 20 to 24)	\$2,300	\$27,600	68.6%
15	EC2-2	Developer instances off outside working hours	Kofi Brandvold	tp-dev	week 3 (April 20 to 24)	\$2,258	\$27,095	71.2%
16	EKS-1	Karpenter consolidation, production nodes	Kofi Brandvold	tp-prod	week 4 (April 27 to May 1)	\$2,243	\$26,911	73.8%
17	NET-1	S3 and DynamoDB gateway endpoints	Kofi Brandvold	tp-prod	week 3 (April 20 to 24)	\$2,230	\$26,763	76.3%
18	KIN-2	Kinesis from 100 shards to 20	Lior Haddad	tp-prod	week 4 (April 27 to May 1)	\$2,150	\$25,795	78.8%
19	DDB-1	Provisioned capacity on the two steady tables	Lior Haddad	tp-prod	week 4 (April 27 to May 1)	\$2,088	\$25,058	81.1%
20	SP-1	Compute Savings Plan	Gwen Tallis	tp-management	week 4 (April 27 to May 1)	\$1,980	\$23,764	83.4%
21	EC2-5	Transcoders scale on queue depth	Kofi Brandvold	tp-prod	week 4 (April 27 to May 1)	\$1,647	\$19,763	85.3%
22	ATH-1	Athena on partitioned Parquet	Lior Haddad	tp-prod	week 5 (May 4 to 8)	\$1,445	\$17,340	86.9%
23	RDS-2	Writer down one size	Dev Raman	tp-prod	week 5 (May 4 to 8)	\$1,313	\$15,759	88.4%
24	EC2-3	API tier minimum from 9 to 6	Marcus Oyelaran	tp-prod	week 3 (April 20 to 24)	\$993	\$11,914	89.5%
25	EC2-1b	Staging on weekdays only	Kofi Brandvold	tp-staging	when its condition is met; the owner sets the date at the readout	\$935	\$11,218	90.6%
26	SP-2	SageMaker Savings Plan	Gwen Tallis	tp-management	after SM-1a, on the hosting floor it leaves	\$933	\$11,192	91.7%

Rank	ID	Change	Owner	Account	Proposed for	A month	A year	Cumulative share
27	KIN-1	Kinesis retention from 365 days to 7	Lior Haddad	tp-prod	week 3 (April 20 to 24)	\$920	\$11,040	92.7%
28	EBS-2	Unattached volumes archived, then deleted	Kofi Brandvold	tp-prod	week 3 (April 20 to 24)	\$806	\$9,677	93.6%
29	OS-1	OpenSearch data nodes down one size	Lior Haddad	tp-prod	week 5 (May 4 to 8)	\$731	\$8,778	94.5%
30	RDS-1	Read replicas down one size	Dev Raman	tp-prod	week 4 (April 27 to May 1)	\$656	\$7,866	95.2%
31	EBS-1	gp2 to gp3	Kofi Brandvold	tp-prod	week 3 (April 20 to 24)	\$616	\$7,397	95.9%
32	SM-3	Moderation classifier to Inferentia2	Yuki Sorensen	tp-prod	when its condition is met; the owner sets the date at the readout	\$610	\$7,323	96.6%
33	SM-1b	Embedding endpoint scales on its backlog	Yuki Sorensen	tp-prod	week 5 (May 4 to 8)	\$553	\$6,636	97.2%
34	EC2-4	API tier on Graviton	Marcus Oyelaran	tp-prod	week 4 (April 27 to May 1)	\$511	\$6,132	97.8%
35	CACHE-2	Redis OSS to Valkey	Kofi Brandvold	tp-prod	when its condition is met; the owner sets the date at the readout	\$479	\$5,754	98.3%
36	ECR-1	ECR lifecycle policy	Kofi Brandvold	tp-prod	week 3 (April 20 to 24)	\$440	\$5,284	98.9%
37	TR-1	Transcribe only videos with speech	Amara Delacroix	tp-prod	week 3 (April 20 to 24)	\$395	\$4,740	99.3%
38	S3-4	Old exports and CI artifacts expired	Kofi Brandvold	tp-prod	week 3 (April 20 to 24)	\$344	\$4,129	99.7%
39	CDN-1b	One rendition fewer to transcode	Kofi Brandvold	tp-prod	week 5 (May 4 to 8)	\$220	\$2,635	99.9%
40	NET-2	Public IPv4 addresses from 20 to 6	Kofi Brandvold	tp-prod	week 3 (April 20 to 24)	\$51	\$613	100.0%

Rank	ID	Change	Owner	Account	Proposed for	A month	A year	Cumulative share
	Total					\$87,867	\$1,054,404	100.0%

Who owns the changes

The owner of each change makes the decision about it. Noor Castellanos own none and consult on the rest.

Owner	Title	Changes	A yearIDs
Dev Raman	Co-founder and Chief Technology Officer	2	\$23,625RDS-1, RDS-2
Kofi Brandvold	Head of Platform	15	\$263,424EC2-1a, EC2-1b, EC2-2, EC2-5, EKS-1, S3-4, NET-1, NET-2, CDN-1b, EBS-1, EBS-2, EBS-3, ECR-1, CACHE-1, CACHE-2
Sasha Ilyin	Head of Security and Compliance	1	\$32,197LOG-2
Yuki Sorensen	Head of Machine Learning	4	\$172,899SM-1a, SM-1b, SM-2, SM-3
Lior Haddad	Head of Data	6	\$118,651DDB-1, KIN-1, KIN-2, S3-1, ATH-1, OS-1
Amara Delacroix	Head of Trust and Safety	5	\$276,843MOD-1, MOD-2, TR-1, S3-2, S3-3
Bea Okafor	Mobile Engineering Lead	1	\$55,443CDN-1
Marcus Oyelaran	Backend Lead	4	\$76,366EC2-3, EC2-4, LOG-1, LOG-3
Gwen Tallis	Vice President, Finance	2	\$34,956SP-1, SP-2

Which change closes which HRI

The Well-Architected Remediation accepted five high-risk issues (HRIs) in writing in March, because the fix for each is a cost change, and a change is counted by exactly one engagement. Of the 44 entries on the shared ledger, 40 are this engagement's. Each HRI below closes when the last change on its row is in place.

The five accepted HRIs. SM-1a and SM-2 close two each: COST06 and SUS02.

HRI	Question	Signed by	Risk while it waits	Closed by	Last change
PERF03	How do you store, manage, and access data in your workload?	Lior Haddad, March 9, 2026	Each Athena query keeps scanning raw JSON.	S3-1, ATH-1	S3-1, proposed for week 5 (May 4 to 8)
COST06	How do you meet cost targets when you select resource type, size and number?	Gwen Tallis, March 10, 2026	Oversized compute keeps running at its current size.	EC2-2, SM-1a, SM-2	SM-1a, proposed for week 4 (April 27 to May 1)
COST07	How do you use pricing models to reduce cost?	Gwen Tallis, March 10, 2026	Everything keeps billing at On-Demand rates.	SP-1	SP-1, proposed for week 4 (April 27 to May 1)
SUS02	How do you align cloud resources to your demand?	Yuki Sorensen, March 11, 2026	GPU endpoints and notebooks keep running around the clock.	SM-1a, SM-2	SM-1a, proposed for week 4 (April 27 to May 1)
SUS04	How do you take advantage of data management policies and patterns to support your sustainability goals?	Lior Haddad, March 9, 2026	Logs, the telemetry stream, and raw telemetry keep growing without limit.	KIN-1, LOG-2, S3-1, S3-3	S3-1, proposed for week 5 (May 4 to 8)

One-time charges

Four changes carry a one-time charge, named here and never annualized into a saving.

One-time charges, at estimate.

Change	What the charge is	Amount
S3-1: Telemetry to Parquet, raw JSON to Deep Archive	Glue conversion, and the Deep Archive transition requests	\$461
S3-3: Renditions over 60 days to Glacier Instant Retrieval	Glacier Instant Retrieval transition requests	\$2,560
LOG-2: Application logs kept 30 days	The export's writes to S3 Standard, its day there before the lifecycle rule moves it, and the Deep Archive transition requests	\$141
S3-2: Originals over 90 days to Glacier Instant Retrieval	Glacier Instant Retrieval transition requests, when the rule lands	\$192
All four		\$3,354

What is not counted

The roadmap counts the 40 changes above and nothing else. These dollars are real, and they are left out on purpose.

- **Cost avoidance on growth.** Tidepost plans for 1,500,000 monthly users by December 2026. The waste this report removes grows with users if nothing stops it, and the guardrails stop it: five the Optimization builds in week five, and two already in place from earlier engagements (GR-5 and GR-7). Avoiding that is worth about \$140,000 to \$230,000 a year at the December run rate. It is **cost avoidance, not counted**, because it is a forecast and not a bill.
- **The CloudFront private pricing negotiation.** CloudFront at \$25,488 a month is large enough to negotiate. That is Gwen Tallis's, it is named in the roadmap, and it is not priced and not counted, since there is no offer to price.
- **The fall in Business Support+.** Support is tiered, and each dollar of usage removed takes 5% of a dollar off it. That is a consequence of the savings, not a change, and no report counts it.
- **One-time charges.** The \$3,354 above is shown beside the savings and is not annualized into them.
- **What earlier engagements count.** The forgotten GPU instance and three Well-Architected changes (the ranking endpoint, the PostgreSQL 16 upgrade, and the production EKS upgrade) are on the ledger under their own engagements.

- **Considered and not priced.** Spot for the g4dn, 2xlarge transcoders and SageMaker managed Spot training: Spot prices move with demand, so no table prices a saving, and no figure here depends on either.
- **The Savings Plans’ discount in any other change.** Every other change is priced on public On-Demand cost, so the plans are counted once, in SP-1 and SP-2.

Dates that matter

These are not savings. They are dates after which a version costs more or stops being supported.

Engine versions and support dates. The staging rebuild (EC2-1a) retires the cluster in extended support before its date, and the development cluster needs its own upgrade.

What	Version	Support
RDS for PostgreSQL 13 (production until March 22; staging until its retirement)	PostgreSQL 13	Standard support ended February 28, 2026; RDS Extended Support bills from March 1, 2026
RDS for PostgreSQL 16 (production from March 22)	PostgreSQL 16	Standard support to February 28, 2029
Amazon EKS, production and staging	Kubernetes 1.30	Standard support ended July 23, 2025; extended support until July 23, 2026
Amazon EKS, development	Kubernetes 1.33	Standard support to July 29, 2026
Amazon ElastiCache	Redis OSS 7.1	Supported; Valkey 8 is the same API at a lower node price (CACHE-2)
Amazon OpenSearch Service	OpenSearch 2.13	Supported

What the Analysis could not see

- **Peaks.** Utilization comes from CloudWatch over the weeks the data holds, and a peak the data does not hold would change a sizing change. Each card names the evidence and the owner who can confirm it.
- **Content.** No one at Vantalect reads the contents of any bucket, table, or log message.
- **A change’s effect on users.** MOD-1, MOD-2, and CDN-1 change what users and moderators see. Their prices are exact; their effects are the owners’ to judge.

AWS, Amazon Web Services, AWS Graviton, AWS Inferentia, and the AWS service names are trademarks of Amazon.com, Inc. or its affiliates. PostgreSQL is a registered trademark of the PostgreSQL Community Association of Canada. Kubernetes is a registered trademark of The Linux Foundation. Redis is a registered trademark of Redis Ltd. Valkey is a trademark of the Linux Foundation. OpenSearch is a registered trademark of Amazon.com, Inc. SOC 2 is a registered trademark of the American Institute of Certified Public Accountants. CIS Benchmarks is a trademark of the Center for Internet Security, Inc. GitHub is a trademark of GitHub, Inc. Jupyter is a trademark of LF Charities. Vantalect is not affiliated with or endorsed by any of them. Tidepost, Inc. is fictional.