

AWS COST · SAMPLE DELIVERABLE

Measured on the Cost and Usage Report

Week 6 of 6: Measure · prepared for Tidepost, Inc.

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

May 15, 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

Result	2
The method	4
The query for one change	7
The Savings Plan, by its ARN.....	8
Measured against estimated	9
Every change	9
Where the measurement fell short, and where it did not	11
The gate	14
Robustness	15
One-time charges.....	15
Approved, then declined	15
What is not counted	16
Re-running this measurement	16

Result

The 31 changes in place cut \$725,391 a year from Tidepost’s AWS bill, **34.4% of the April run rate (\$2,110,762 a year), measured on Tidepost’s own Cost and Usage Report.** That is 92.9% of the \$780,703 the Analysis estimated for the same changes. Tidepost has no private pricing agreement, so every figure here is at AWS list prices, which is what its invoice falls by at April usage. Counting the one change approved at the readout and then declined, as the engagement’s terms define it, the figure for the gate is **\$736,583 a year** against a gate of \$105,000, three times the Optimization’s \$35,000 list fee. That is 7.0 times the gate, with \$631,583 to spare, so the second invoice, \$14,875, is issued with this report.

Without the declined change the margin is \$620,391, so the gate does not depend on it. The 15% bundle discount set Tidepost’s fee at \$29,750, and it does not move the gate. The measured savings repay that fee in about 15 days, and the first year nets \$695,641 after it; all four of this year’s Vantalect fees (\$114,250) take longer, because the earlier engagements added to the bill: security services (\$2,584 a month) and the Well-Architected Remediation (\$10,094 a month), less the forgotten GPU instance the security engagement removed (\$4,141 a month), a net \$8,537 a

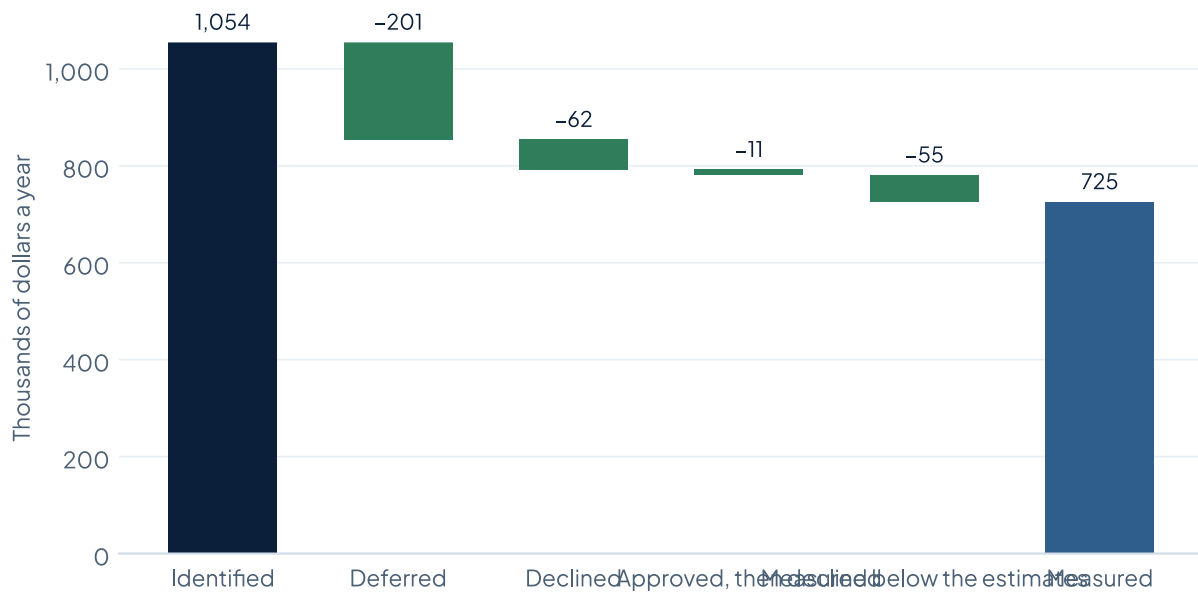
month. Net of that, the measured savings repay all four fees in about 67 days. The counted figure decides the gate and nothing else. The changes' one-time charges, \$3,164, are listed in [the one-time charges](#) and never annualized; with them deducted, the counted figure is \$733,419.

The result against the gate. The full computation is in [the gate computation](#).

The result, at list prices	Amount
Measured savings, 31 changes in place, annualized	\$725,391
Share of the April run rate (\$2,110,762 a year)	34.4%
Share of the Analysis's estimates for the same changes	92.9%
Counted for the gate, with SP-2 at its estimate	\$736,583
The gate: three times the \$35,000 list fee	\$105,000
Margin over the gate	\$631,583
Monthly bill reduction at April usage, measured	\$60,449

The second invoice

The Optimization's second \$14,875 is invoiced only if the annualized savings in place at the end of week six reach three times the fee. They reach \$736,583, so the invoice is issued with this report, on May 15, 2026. If you disagree with any figure here, the queries in [the queries](#) reproduce it from Tidepost's own data in the `cost-analysis` workgroup.



From what the Analysis identified to what the Cost and Usage Report measured. Every step is a row in [the gate computation](#).

The method

The terms of the engagement say how the second half of the fee is decided:

The Optimization is invoiced in full only if the annualized savings in place at the end of week six are at least three times the fee. Savings are measured change by change on the client's Cost and Usage Report: the cost of each changed resource in the week before the change against the week after it, and the on-demand cost of the usage each commitment covers less its amortized cost, annualized. A change the client approved at the week-two readout and later declines counts at the savings the roadmap estimated for it.

In a bundle the gate stays at three times the Optimization's list fee, so Tidepost's gate is \$105,000. Each step below applies that clause to Tidepost's bill, the same way for every change.

1. The cost. Every change except SP-1 is measured on **public On-Demand cost**:

pricing_public_on_demand_cost on usage a Savings Plan covered, and the unblended cost elsewhere, which for Tidepost is the same list price. The source is the CUR 2.0 data export tidepost-cur2 in tp-management, which has run since January 28, 2026. Measuring on public On-Demand cost keeps SP-1 out of every other change's cost, so the plan is counted once, as its own change. Four changes to EC2 usage (EKS-1, EC2-5, EC2-4, CDN-1b) have after-windows that run past SP-1's purchase on April 30; on this basis their savings are what the usage would cost without the plan, and the plan's discount stays in SP-1.

2. The window. For each changed resource, take its cost for each of the seven days before the change and the seven days after it. The day of the change is excluded from both, because a change lands partway through its day. Each window is summed and divided by its number of days, not averaged over the rows present: a deleted resource has no rows after the change, and a day with no rows is a day with no cost. The daily saving is the mean before less the mean after, and the annual saving is that difference times 365. Where two changes touch the same usage on the same resource less than fifteen days apart, a full week each way would put one change's day inside the other's window, so the days between the two change days are split: the first change's after-window takes the first half and the second change's before-window the second, so the two windows share no day and neither holds the other change. Each of those changes uses windows of that length on both sides: SM-1a and SM-1b, the embedding endpoint's instance-hours, three days each (SM-1a's after-window April 28 to April 30, SM-1b's before-window May 1 to May 3); EC2-3 and EC2-4, the API tier's instance-hours, four days each (EC2-3's after-window April 23 to April 26, EC2-4's before-window April 27 to April 30); EC2-5 and CDN-1b, the transcoder fleet's instance-hours, three days each (EC2-5's after-window April 29 to May 1, CDN-1b's before-window May 3 to May 5). KIN-1 and KIN-2 change one stream but different usage types (long-term retention against shard-hours and PUT units), so each keeps its week. Every row is the day's own usage at list price: a store that grows with ingestion (LOG-2 and KIN-1) is read at its size on each day, and log events CloudWatch Logs had not yet deleted (LOG-2) are on the days they were billed.

3. What is in the cost. Only the line types Usage and SavingsPlanCoveredUsage. Fees, taxes, credits, and refunds are not usage a change moves. Business Support+ bills monthly, so it is in no daily series, and its charges carry no tags; it is excluded by product code.

4. The lag. The Cost and Usage Report finalizes a day's rows about 48 hours later. This measurement ran on Friday, May 15, on data through May 13. A change needs seven full days after it, so only changes made by May 6 could be measured; every counted change was in place between April 21 and May 6.

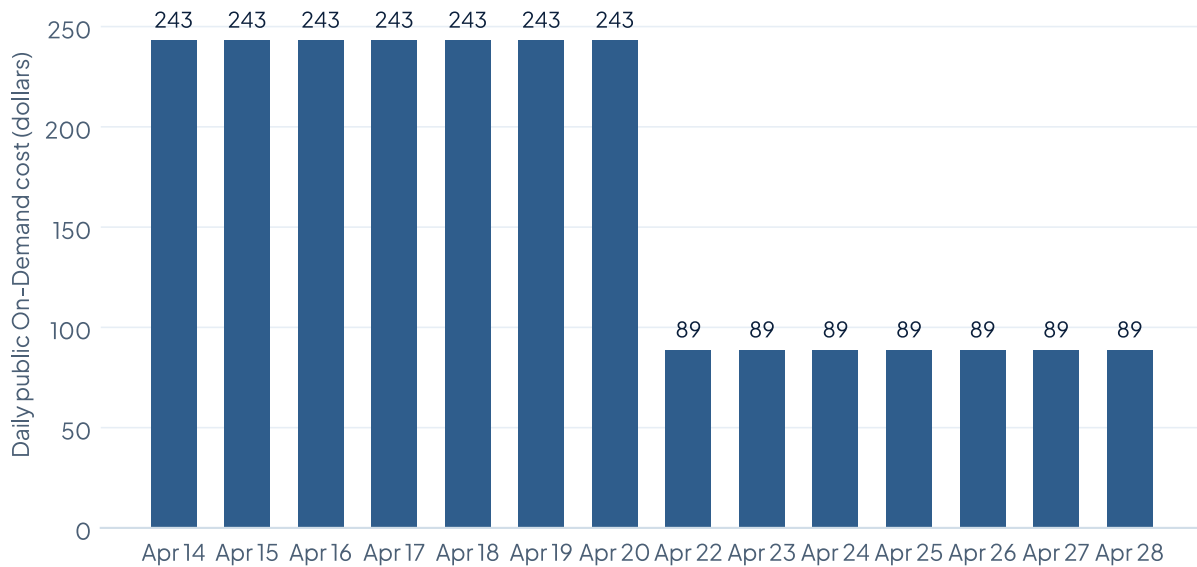
5. **The Savings Plan (SP-1)** is measured by its ARN, not by a resource: the On-Demand cost of the usage it covered, less its fee, with its utilization ([the Savings Plan query](#)).
6. **S3 changes (S3-1, S3-3, S3-4)** are attributed by prefix and storage class from S3 Inventory and Storage Lens. Telemetry, renditions, and exports are separate prefixes, so no two S3 changes share a measurement. **LOG-2** is measured on the application log groups' CloudWatch Logs storage, by log group, plus the S3 Glacier Deep Archive storage of the counsel export prefix in tp-log-archive on each after-day, because the change could not happen without it. The export tasks wrote to S3 Standard, and a lifecycle rule moved each object to Deep Archive the next day; those writes, that day in Standard, and the transition requests are LOG-2's one-time charge.
7. **CDN-1 is normalized to bytes per view.** Viewing grows week to week; the measurement compares CloudFront cost per view in the two windows, times the views of the before-window, so traffic growth is never read as a saving.
8. **One-time charges** are listed beside the result and never annualized into it.
9. **Nothing is counted that is not in place.** Deferred and declined changes count for nothing. The one exception is the rule the terms state for a change approved at the readout and declined later ([SP-2](#)).

How each counted change was measured.

Measured by	ChangesWhich
Public On-Demand cost of the changed resources	25SM-1a, MOD-1, SM-2, EBS-3, LOG-1, CACHE-1, LOG-3, NET-1, KIN-2, DDB-1, EC2-2, EKS-1, EC2-5, ATH-1, KIN-1, EC2-3, EBS-2, RDS-1, EBS-1, EC2-4, ECR-1, TR-1, SM-1b, CDN-1b, NET-2
Public On-Demand cost per view	1CDN-1
Public On-Demand cost, CloudWatch Logs storage by log group and the export prefix by storage class	1LOG-2
Public On-Demand cost, S3 by prefix and storage class	3S3-1, S3-3, S3-4
Savings Plan ARN	1SP-1
Total	31

A worked example

SM-2, notebook auto-stop after 60 idle minutes in tp-sandbox-ml, went in on Tuesday, April 21. The seven daily rows before it (April 14 to April 20) average \$243.12 a day for the three notebook instances. The seven after it (April 22 to April 28) average \$88.66. The daily saving is \$154.46; times 365 is \$56,379. The Analysis estimated \$61,266 (1 ml.g5.12xlarge at \$7.09 and 2 ml.g5.2xlarge at \$1.52, from 730 hours to 220 and 240), so SM-2 measures at 92.0% of its estimate: the notebooks ran 262 and 276 hours a month, not 220 and 240.



SM-2's daily public On-Demand cost in the two windows. April 21, the day of the change, is in neither.

The query for one change

Anyone on Tidepost's team can run the measurement for one change in Athena, in the cost-analysis workgroup in tp-management, against the table `tidepost_cur.cur2`. The query below is SM-2's. For another change, replace the account, service, and usage filter (or filter on a cost allocation tag: user-defined keys appear in the `resource_tags` map with a `user_prefix`, as in `element_at(resource_tags, 'user_team') = 'ml'`), the dates, and the billing period.

SQL

```
WITH daily AS (
    SELECT date(line_item_usage_start_date) AS day,
           sum(CASE WHEN line_item_line_item_type = 'SavingsPlanCoveredUsage' THEN
pricing_public_on_demand_cost
                ELSE line_item_unblended_cost
            END) AS cost
    FROM tidepost_cur.cur2
    WHERE line_item_usage_account_name = 'tp-sandbox-ml'
        AND product_servicecode = 'AmazonSageMaker'
        AND line_item_usage_type LIKE '%Notebk:ml.g5%'
        AND line_item_line_item_type IN ('Usage', 'SavingsPlanCoveredUsage')
        AND billing_period IN ('2026-04')
    GROUP BY 1
), windows AS (
    SELECT sum(CASE WHEN day BETWEEN date '2026-04-14' AND date '2026-04-20' THEN cost ELSE 0
END) / 7.0 AS daily_before,
           sum(CASE WHEN day BETWEEN date '2026-04-22' AND date '2026-04-28' THEN cost ELSE 0
END) / 7.0 AS daily_after
    FROM daily
)
SELECT daily_before, daily_after, (daily_before - daily_after) * 365 AS annual_saving
FROM windows;
```

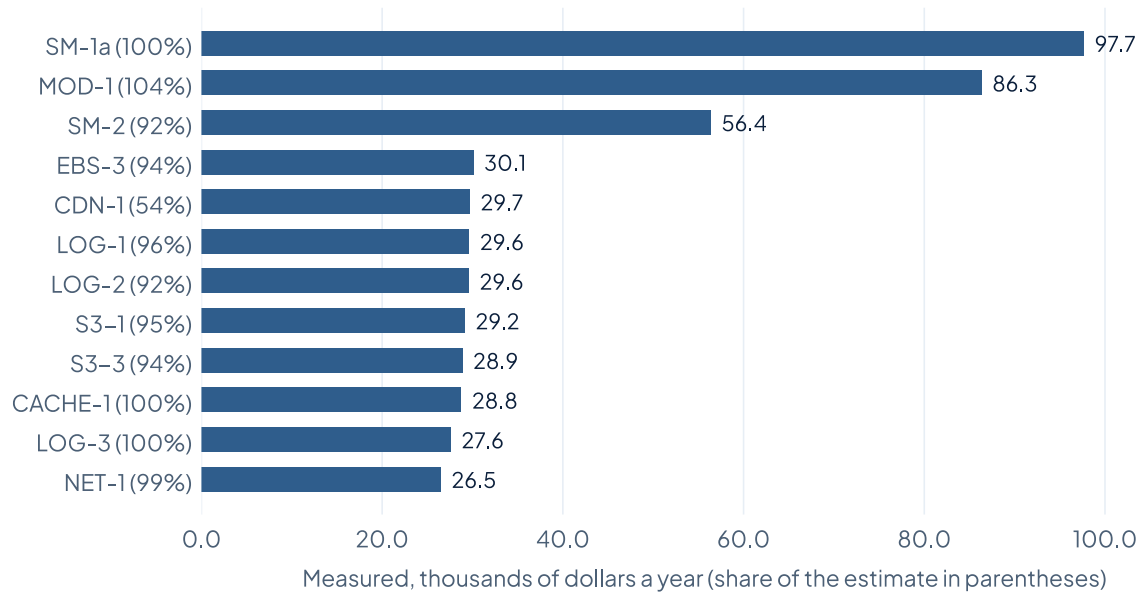
The Savings Plan, by its ARN

SP-1 is a one-year, No Upfront Compute Savings Plan at \$7.27 an hour, bought in tp-management on April 30 with sharing on, so it applies across the organization. Its saving is the On-Demand cost of the usage it covered less its fee, read from the lines that carry its ARN. In the seven days from May 1 to May 7 it covered \$239.59 a day of On-Demand usage (\$9.9828 an hour) for a fee of \$174.48 a day, 27.2% off, and it was fully used every hour, because the always-on compute floor of \$14.26 an hour of On-Demand usage is above what it covers. Annualized, that is \$23,764, its estimate: a fully used plan bought on a known floor saves exactly its discount, which is why it matches.

SQL

```
SELECT date(line_item_usage_start_date) AS day,
       sum(CASE WHEN line_item_line_item_type = 'SavingsPlanCoveredUsage' THEN
pricing_public_on_demand_cost ELSE 0 END) AS covered_on_demand,
       sum(CASE WHEN line_item_line_item_type = 'SavingsPlanRecurringFee' THEN
line_item_unblended_cost ELSE 0 END) AS fee,
       sum(CASE WHEN line_item_line_item_type = 'SavingsPlanRecurringFee' THEN
savings_plan_used_commitment ELSE 0 END)
       / nullif(sum(CASE WHEN line_item_line_item_type = 'SavingsPlanRecurringFee' THEN
savings_plan_total_commitment_to_date ELSE 0 END), 0) AS utilization
    FROM tidepost_cur.cur2
    WHERE savings_plan_savings_plan_a_r_n = '<the SP-1 ARN from the Savings Plans console>'
        AND date(line_item_usage_start_date) BETWEEN date '2026-05-01' AND date '2026-05-07'
    GROUP BY 1
    ORDER BY 1;
```

Measured against estimated



The twelve largest measured savings, with each one's share of its estimate. Every change is in [the table below](#).

Every change

Each in-place change, largest measured saving first. The reason each one differs from its estimate, or matches it, is in [the next section](#).

Id	Change	In place	Estimate a year	Measured a year	Of estimate
SM-1a	Embedding endpoint from ml.g5.12xlarge (one GPU of four used) to ml.g5.2xlarge	Apr 27	\$97,674	\$97,674	100.0%
MOD-1	Rekognition Video only for accounts under 90 days old and for reported videos; the frame path covers the rest	Apr 28	\$83,160	\$86,344	103.8%
SM-2	Notebook auto-stop after 60 idle minutes	Apr 21	\$61,266	\$56,379	92.0%
EBS-3	The cron's snapshot chains past 35 days deleted; AWS Backup governs from here	May 4	\$31,949	\$30,106	94.2%
CDN-1	Rendition ladder: 720p on cellular, no 1080p for vertical clips (app release April 29)	Apr 29	\$55,443	\$29,708	53.6%

Id	Change	In place	Estimate a year	Measured a year	Of estimate
LOG-1	API log level from DEBUG to INFO in production	Apr 22	\$30,720	\$29,644	96.5%
LOG-2	Application log groups kept 30 days; January to March exported for counsel first and kept in S3 Glacier Deep Archive	May 5	\$32,197	\$29,605	91.9%
S3-1	Telemetry to Parquet at ingest; 151 TB of raw JSON converted, then moved to S3 Glacier Deep Archive	May 6	\$30,640	\$29,155	95.2%
S3-3	Renditions older than 60 days to S3 Glacier Instant Retrieval (priced against Standard-IA and Intelligent-Tiering on the same inventory)	May 5	\$30,734	\$28,891	94.0%
CACHE-1	ElastiCache nodes from cache.r6g.4xlarge to cache.r6g.2xlarge (peak 29 GB used of 105 a shard)	Apr 28	\$28,768	\$28,768	100.0%
LOG-3	Custom metrics from 12,000 to 3,000 (the per-post dimension dropped)	Apr 22	\$27,600	\$27,600	100.0%
NET-1	S3 and DynamoDB gateway endpoints	Apr 21	\$26,763	\$26,497	99.0%
KIN-2	Kinesis from 100 shards to 20, records aggregated in the collector	Apr 30	\$25,795	\$25,809	100.1%
SP-1	Compute Savings Plan, one year, No Upfront, \$7.27 an hour (70% of the floor's plan cost), bought in tp-management with sharing on	Apr 30	\$23,764	\$23,764	100.0%
DDB-1	Provisioned capacity with auto scaling on the two steady tables (85% of the traffic)	Apr 30	\$25,058	\$23,355	93.2%
EC2-2	Developer instances stop outside working hours	Apr 21	\$27,095	\$22,534	83.2%
EKS-1	Karpenter consolidation of the production nodes (12 to 8)	Apr 27	\$26,911	\$20,183	75.0%
EC2-5	Transcoders scale on queue depth (6 always on to an average of 3)	Apr 28	\$19,763	\$18,106	91.6%
ATH-1	Athena on partitioned Parquet	May 6	\$17,340	\$16,619	95.8%
KIN-1	Kinesis retention from 365 days to 7 (the raw stream is already in S3)	Apr 24	\$11,040	\$11,992	108.6%

Id	Change	In place	Estimate a year	Measured a year	Of estimate
EC2-3	API tier minimum from 9 to 6 (two a zone), with target tracking	Apr 22	\$11,914	\$9,844	82.6%
EBS-2	Unattached volumes archived as snapshots, then deleted (9 TB)	Apr 23	\$9,677	\$9,677	100.0%
RDS-1	Read replicas from db.r6g.2xlarge to db.r6g.xlarge (working set 21 GB)	Apr 30	\$7,866	\$7,866	100.0%
EBS-1	gp2 to gp3 (30.1 TB)	Apr 22	\$7,397	\$7,397	100.0%
EC2-4	API tier on Graviton (c7g.4xlarge)	May 1	\$6,132	\$6,855	111.8%
ECR-1	ECR lifecycle policy (5 TB to 0.7 TB)	Apr 23	\$5,284	\$5,284	100.0%
TR-1	Transcribe only videos with detected speech	Apr 24	\$4,740	\$4,366	92.1%
S3-4	Old exports and CI artifacts expired (16 TB)	Apr 23	\$4,129	\$4,129	100.0%
SM-1b	Embedding endpoint scales on its backlog, one to four instances	May 4	\$6,636	\$3,981	60.0%
CDN-1b	One rendition fewer to transcode	May 6	\$2,635	\$2,646	100.4%
NET-2	Public IPv4 addresses from 20 to 6	Apr 21	\$613	\$613	100.0%
Total	31 changes		\$780,703	\$725,391	92.9%

Where the measurement fell short, and where it did not

A measurement that matched every estimate would be a model, not a measurement. Fifteen changes measured more than 1.5% below their estimates, three measured more than 1.5% above, ten matched exactly, and the other three landed within 1.5% either way. Each reason is the physical fact that moved it, and where day-to-day usage in the two windows moved a figure by half a point or more, the reason says by how much.

The changes that measured below their estimates, the largest gap first.

Id	Estimate	Measured	GapWhy
CDN-1	\$55,443	\$29,708	-\$25,73652% of viewers had the new app in the window; bytes per view normalized; day-to-day usage in the two windows moved it +1.6 points from the 52.0% those alone give
EKS-1	\$26,911	\$20,183	-\$6,728The cluster settled at 9 nodes, not 8: two pods with zone constraints
SM-2	\$61,266	\$56,379	-\$4,887The notebooks ran 262 and 276 hours a month, not 220 and 240
EC2-2	\$27,095	\$22,534	-\$4,5612 of the 12 engineers took the exemption tag and ran around the clock
SM-1b	\$6,636	\$3,981	-\$2,654A breaking-news spike on May 5 held 2.6 instances that day, so the endpoint averaged 1.7 over its after-window, not 1.5
LOG-2	\$32,197	\$29,605	-\$2,592The log groups held 94.0 TB on the change day, not the 89.8 of April 6, because they grew with ingestion; CloudWatch Logs took three days to delete the expired events (62% of them were still billed on the first day after and 21% on the second); and the export held 23 TB, not 20
EC2-3	\$11,914	\$9,844	-\$2,070The tier averaged 7.4 instances, not 7; day-to-day usage in the two windows moved it +2.6 points from the 80.0% those alone give
EBS-3	\$31,949	\$30,106	-\$1,84349 TB of unique blocks freed, not 52: three volumes share blocks with kept snapshots
S3-3	\$30,734	\$28,891	-\$1,843A story from August 2025 resurfaced on May 9 and pulled 15 TB back, not 10, at the Glacier Instant Retrieval retrieval charge
DDB-1	\$25,058	\$23,355	-\$1,703Auto scaling held 155% of the mean during the May 5 spike, not 140%
EC2-5	\$19,763	\$18,106	-\$1,657The fleet averaged 3.2 instances, not 3; day-to-day usage in the two windows moved it -1.7 points from the 93.3% those alone give
S3-1	\$30,640	\$29,155	-\$1,4854% of raw partitions failed conversion and stayed in Standard until a rerun
LOG-1	\$30,720	\$29,644	-\$1,076Ingestion settled at 1.8 TB a month, not 1.6
ATH-1	\$17,340	\$16,619	-\$721Analysts scanned 50 TB, not 41; day-to-day usage in the two windows moved it -1.0 points from the 96.9% those alone give

Id	Estimate	Measured	GapWhy
TR-1	\$4,740	\$4,366	-\$37435% of videos had no speech in the after-window, not 38%

MOD-1 measured above its estimate, \$86,344 against \$83,160 (103.8%): 42% of videos came from older accounts and went unreported in the after-window, not 40%; day-to-day usage in the two windows moved it -1.2 points from the 105.0% those alone give. **KIN-1 measured above its estimate**, \$11,992 against \$11,040 (108.6%): the long-term store grew by a day's ingest, about 246 GB, every day until the change, so its before-window held 43,448 GB on average, not the 40,000 of April 6; the line stopped when the data aged out. **EC2-4 measured above its estimate**, \$6,855 against \$6,132 (111.8%): the saving scales with the 7.4 instances that ran, not 7; day-to-day usage in the two windows moved it +6.1 points from the 105.7% those alone give.

The changes that landed within 1.5% of their estimates without matching exactly, and what moved each one.

Id	Estimate	Measured	GapWhy
NET-1	\$26,763	\$26,497	-\$26679% of NAT bytes moved, not 80%: a vendor SDK writes to its own bucket in us-west-2, and a gateway endpoint carries only same-Region traffic
CDN-1b	\$2,635	\$2,646	\$11The fleet averaged 3.2 instances, not 3, so the 13% it skips is 0.43 of an instance, not 0.4; day-to-day usage in the two windows moved it -6.2 points from the 106.7% those alone give
KIN-2	\$25,795	\$25,809	\$14Shard-hours bill the same every day; the PUT units (\$109.96 a month before, \$4.40 after aggregation) move with traffic day to day

CDN-1 has the largest gap, and it is not a miss. The rendition ladder ships in the mobile app, and only the share of viewers on the new version can benefit from it. As more viewers update, the saving grows toward its estimate, but this report counts only what the week after April 29 measured.

The exact matches. Each prices a fixed quantity that did not move between the two windows.

Id	Measured	Why it matches its estimate exactly
SM-1a	\$97,674	A fixed hourly price on fixed instance-hours: the measurement matches the estimate exactly
CACHE-1	\$28,768	Hourly: the measurement matches exactly
LOG-3	\$27,600	Metrics bill by the hours they exist: the measurement matches exactly
SP-1	\$23,764	Fully used every hour after April 30: the On-Demand cost it covered less its fee, so it matches its estimate
EBS-2	\$9,677	Fixed sizes: the measurement matches exactly
RDS-1	\$7,866	Hourly: the measurement matches exactly
EBS-1	\$7,397	Same size, new price: the measurement matches exactly
ECR-1	\$5,284	Storage: the measurement matches exactly
S3-4	\$4,129	Storage stops the day objects expire: the measurement matches exactly
NET-2	\$613	Hourly: the measurement matches exactly

The gate

The gate computation. In place, deferred, declined, and approved-then-declined together equal the identified figure to the dollar.

Step	A year
Identified by the Analysis (37 findings, 40 changes)	\$1,054,404
Less deferred (5 changes)	-\$200,550
Less declined at the readout (3 changes)	-\$61,959
Less approved, then declined (SP-2)	-\$11,192
In place at the estimates (31 changes)	\$780,703
Less measured below the estimates, net	-\$55,312
Measured on the Cost and Usage Report	\$725,391
Plus SP-2 at its estimate, as the terms require	\$11,192
Counted for the gate	\$736,583

Step	A year
The gate	\$105,000
Margin over the gate	\$631,583

Robustness

The 21 changes whose usage moves day to day each carry sampling error from their daily rows. Taken together, two standard errors are \$16,715 a year, so the measured figure lies between \$708,676 and \$742,106; even the lower bound clears the gate by \$603,676. The other 10 changes price fixed quantities and carry no sampling error. Without the three largest changes (SM-1a, MOD-1, SM-2), the measured savings would still be \$484,994, and without SP-2 the margin is \$620,391. No single change, and no week's noise, decides the gate.

One-time charges

One-time charges, never annualized. S3-1's is Glue conversion and the S3 Glacier Deep Archive transition requests; S3-3's is the Glacier Instant Retrieval transition requests; LOG-2's is the export's writes to S3 Standard, its day there, and the Deep Archive transition requests; S3-2's transition requests land with it.

Change	One-time charge	When	In this report's result
S3-1	\$461	May 6	Deducted only in the net-of-one-time figure
S3-3	\$2,560	May 5	Deducted only in the net-of-one-time figure
LOG-2	\$143	May 5	Deducted only in the net-of-one-time figure
S3-2	\$192	May 20	No: it lands after the engagement, with the change

Approved, then declined

SP-2, a one-year SageMaker Savings Plan (\$4.14 an hour covers \$5.4176 of SageMaker hosting at the floor's blended rate (23.6% off)): Approved at the readout, then declined on May 5 by Gwen Tallis: she wants the Inferentia2 move first. The readout was on April 17. The terms count a change approved at the readout and declined later at the savings the roadmap estimated for it, so SP-2 counts at \$11,192. It is the only change counted at an estimate, and the gate is met without it.

What is not counted

- **The 5 deferred changes**, \$200,550 a year at their estimates, count for nothing here. Each has an owner and a date in the handover report.
- **The 3 changes declined at the readout**, \$61,959 a year: MOD-2 (Frame sampling from one frame a second to one every two seconds); RDS-2 (Writer from db.r6g.4xlarge to db.r6g.2xlarge (Multi-AZ)); OS-1 (OpenSearch data nodes from r6g.2xlarge.search to r6g.xlarge.search).
- **What earlier engagements removed**. The forgotten g5.12xlarge (\$4,141 a month) is the security engagement's; the ranking endpoint's move to CPU, production EKS extended support, and PostgreSQL 13 Extended Support on production are the Well-Architected Review's. Each change is counted by exactly one engagement.
- **Cost avoidance, not counted**. The guardrails, five built in week five and two inherited from earlier engagements, keep the waste from growing with users: about \$140,000 to \$230,000 a year at the December run rate. It is labeled and counted for nothing.
- **A CloudFront private pricing negotiation** is Gwen Tallis's to pursue, named in the roadmap, and not counted.

Re-running this measurement

1. Open Athena in tp-management, workgroup cost-analysis, database tidepost_cur; results write to s3://tp-management-athena-results/cost-analysis/.
2. Take the change's resources, account, or tag from its line in the savings roadmap (savings-roadmap.csv), and its date from the change record.
3. Run [the query](#) with the two windows: the seven days before the change day and the seven days after it, or, for two changes to the same usage, the shorter disjoint windows in [step 2](#). Wait until the last day of the after-window is two days old.
4. For a commitment, run [the Savings Plan query](#) on its ARN.

Vantalect read Tidepost's data through vantalect-cost-analysis, which Tidepost deletes on May 15; from then on, Gwen Tallis's team runs these queries with its own access, and nothing in this report depends on a Vantalect login.

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.