

AWS COST · SAMPLE DELIVERABLE

Quick Wins Applied

Week 3 of 6: Quick wins · prepared for Seatlark, Inc.

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

May 29, 2026 · AWS Cost Analysis & Optimization: The Optimization (six weeks)

SAMPLE DELIVERABLE

This is a sample deliverable. Seatlark, Inc. is a fictional company: every account, resource, finding, figure, and person in this report is simulated to show the form and depth of a Vantalect engagement. AWS prices are list prices from the AWS Price List API retrieved on October 13 and 16, 2026, and AWS Business Support+ pricing is from the AWS Support pricing page on October 16, 2026.

Contents

Summary	2
What went in	4
Change by change	5
The waste register	7
How each change was made, and how it is undone	11
Wave 6 and the bill	12
What was declined, and what could still be reversed	13
Honest limits, as of May 29, 2026	14
The exact mechanism	15
The waste runbook	15
Snapshots, volumes, and the schedule	16
Network and logs	17
The migration backups, and verifying on the bill	19
What comes next	19

Summary

In week 3, the four working days after Memorial Day, Seatlark’s engineers put **\$98,760 a year** of savings in place: 14 changes to 159 resources, in waste, schedules, storage, network, logging, and non-production databases. Every change was drafted and reviewed by Vantalect, approved by the owner of the workload it touches, and made on May 27 or May 28. That is 40% of the \$246,839 the Analysis identified, and 94% of the \$105,000 the guarantee asks for ([Week 3 in numbers. Savings are estimates until the week-6 measurement](#)). It does not clear the gate by itself, and it was never going to: the production changes and the commitment come in week 4.

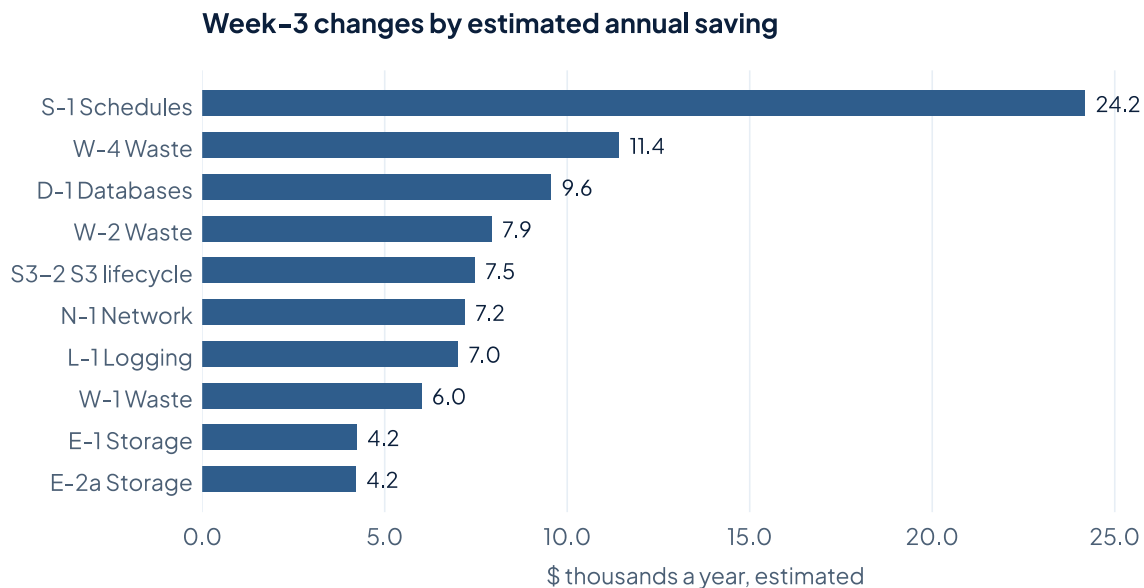
Every dollar in this report is an estimate. It is the Analysis’s price for the change, annualized at AWS list prices. Nothing is measured until week 6, when each change is read off your Cost and Usage Report, and the week-6 figure will differ from this one ([Honest limits, as of May 29, 2026](#) says where we expect it to).

Two things will make the bill harder to read than the savings are. On Saturday, May 30, 2026, wave 6 (partner reporting and the affiliate feed generator) cuts over and adds **\$2,600 a month**. The bill will rise by that much on the same invoice that carries this week’s savings, so it will fall by about

\$5,630 a month, not \$8,230. We measure each change on its own resources, before and after, so growth from the migration is never mistaken for a saving or a loss ([Wave 6 and the bill](#)). And the three changes the owners declined at the readout, N-4, R-3, S-2, count for nothing here and in the gate ([What was declined, and what could still be reversed](#)).

Week 3 in numbers. Savings are estimates until the week-6 measurement.

Week 3 at a glance	Result
Changes put in place	14
Resources changed	159
Annualized savings, estimated	\$98,760
Monthly savings, estimated	\$8,230
One-time cost of making the changes	\$15
Rated High confidence	12 of 14
Made by pull request to your repositories	9
Made by a scripted runbook under a change ticket	5
Week-3 scope declined at the readout (counts for nothing)	\$5,308 a year



The ten largest week-3 changes, estimated and annualized.

Decision for the sponsor

Renee Castellano: no decision is needed to keep this week's savings. Two dates need your yes this week. Confirm June 2, 2026 as the production window for the API tier and the partner APIs with Aaron Feld and Diego Saldana, one working day before the production freeze starts on June 4, 2026, which leaves June 3, 2026 free for a rollback if one is needed. And hold June 3, 2026 for the commitment decision with Lydia Brandt, whom the week-4 report asks to approve a Savings Plan.

What went in

7 kinds of change went in. Waste removal is the largest by count: 4 changes to 31 resources. Schedules are the largest by value, because 11 non-production hosts that ran every hour of the month now run about a third of them.

Week 3 by kind of change. Dollars are estimates at list price. A "resource" is what the change touches: a volume, a server, a log group, a NAT gateway, or a bucket.

Kind of change	Findings	Changes	Resources	Monthly	Annualized	Share of week 3
Waste	W-1, W-2, W-3, W-4	4	31	\$2,144	\$25,733	26%
Schedules	S-1	1	11	\$2,015	\$24,183	24%
Logging	L-1, L-2, L-3	3	8	\$1,089	\$13,064	13%
Network	N-1, N-2	2	14	\$862	\$10,342	10%
Databases	D-1	1	2	\$796	\$9,552	10%
Storage	E-1, E-2a	2	92	\$702	\$8,424	9%
S3 lifecycle	S3-2	1	1	\$622	\$7,462	8%
Total		14	159	\$8,230	\$98,760	100%

12 of the 14 are rated High confidence: the resource is plainly unused, or the change is a configuration with a known price. 2 are Medium: **W-4**, because EBS snapshots are incremental and deleting one frees only the blocks no other snapshot references (the estimate uses each snapshot's billed cost, not its volume size), and **L-2**, because log volume follows traffic. One cost came with the week: **\$15** once, the transition requests into Deep Archive on S3-2. One-time costs are listed again in week 6 with the gate shown net of them, and never netted into the annual savings.

Change by change

Each row names the owner who approved the change, the channel it went through, the date, what the resources were before and after, the evidence that made it safe, and the way back. The "before" and "after" columns count the resources the change touches and price them at list. [The exact mechanism](#) shows how each one was made.

Every week–3 change, in the order it went in: owner, channel, date, before and after, evidence, and rollback.

ID, change, owner, channel, date, saving	Before	After	Evidence	Rollback
W-1 Unattached gp2 volumes from test cutovers. Beth Morrow; change CHG-3122 (scripted runbook); May 27. Saves \$6,000 a year.	10 unattached gp2 volumes, 5,000 GB; \$500 a month	None; \$0 a month	Every volume has been detached 60 to 105 days; CloudWatch shows zero read and write operations; the owning test runbook is closed	Restore from the 30-day snapshot
W-2 MGN replication servers and staging disks left from completed waves. Beth Morrow; change CHG-3123 (scripted runbook); May 27. Saves \$7,927 a year.	14 replication servers and 14 staging disks, 5,600 GB; \$661 a month	None; \$0 a month	All 14 source servers show Cutover complete in MGN; none has replicated in 30 or more days	Re-install the replication agent on the source server (the colo server is retained until the lease ends)
W-3 Two idle load balancers from rehearsals. Aaron Feld; change CHG-3124 (scripted runbook); May 27. Saves \$394 a year.	2 idle Application Load Balancers; \$33 a month	None; \$0 a month	Zero processed bytes and zero healthy targets over 90 days; no Route 53 record points at either	Recreate from the marketplace-infra stack
E-1 gp2 to gp3 on every attached volume. Beth Morrow; change CHG-3121 (scripted runbook); May 27. Saves \$4,224 a year.	17,600 GB on gp2; \$1,760 a month	17,600 GB on gp3, 3,000 IOPS and 125 MiB/s included; \$1,408 a month	gp3 lists at \$0.08 a GB-month with a higher baseline than gp2 for volumes under 1 TB; no volume's measured peak is above 3,000 IOPS	Modify the volume back to gp2 (once in 6 hours)

ID, change, owner, channel, date, saving	Before	After	Evidence	Rollback
D-1 Multi-AZ off on two non-production databases. Beth Morrow; marketplace-infra PR #1808; May 27. Saves \$9,552 a year.	2 databases, Multi-AZ; \$1,592 a month	2 databases, Single-AZ; \$796 a month	Nobody depends on non-production failover; Multi-AZ doubles the instance and storage price	Convert back to Multi-AZ
N-1 S3 gateway endpoints in every VPC. Beth Morrow; seatlark-landing-zone PR #415; May 27. Saves \$7,188 a year.	No S3 gateway endpoint; 28,672 GB a month through 18 NAT gateways; \$1,882 a month	6 endpoints, one per VPC; 15,360 GB a month through NAT; \$1,282 a month	13 TB of the 28 TB the NAT gateways process a month is bound for S3	Remove the endpoint; traffic returns to the NAT gateways
L-1 CloudWatch Logs retention. Beth Morrow; seatlark-landing-zone PR #413; May 27. Saves \$7,004 a year.	5 log groups, none expiring, 22,528 GB stored; \$2,028 a month	Retention 30 or 90 days, 3,072 GB stored; \$1,444 a month	No log group has been queried past 30 days in 90 days of Logs Insights history	Raise the retention
L-3 VPC flow logs to S3. Beth Morrow; seatlark-landing-zone PR #414; May 27. Saves \$2,460 a year.	820 GB a month of flow logs delivered to CloudWatch Logs; \$435 a month	The same logs delivered to S3 with a 30-day lifecycle; \$230 a month	820 GB a month of flow logs	Re-point the flow logs
S3-2 Migration backups to Deep Archive. Owen Pradhan; seatlark-landing-zone PR #416; May 27. Saves \$7,462 a year.	28,672 GB in S3 Standard; \$650 a month	28,672 GB in S3 Glacier Deep Archive; \$28 a month	Written once during the waves; read by nobody since	Restore on request (standard retrieval \$0.02 a GB, 12 hours)
W-4 Snapshot and AMI sprawl. Beth Morrow; change CHG-3125 (scripted runbook); May 28. Saves \$11,412 a year.	30,000 GB of snapshots billed; \$1,581 a month	16,800 GB billed, 5,600 GB of it in the archive tier; \$630 a month	EBS snapshots are incremental, so deleting one frees only the blocks no other snapshot references: the estimate uses each snapshot's billed cost from the resource-level Cost and Usage Report lines, not its volume size	Snapshots in the archive tier restore on request (\$0.03 a GB); deleted snapshots are not recoverable, so each deletion has a signed-off line in the waste register

ID, change, owner, channel, date, saving	Before	After	Evidence	Rollback
E-2a Staging search volumes, io1 to gp3. Imani Brooks; marketplace-infra PR #1809; May 28. Saves \$4,200 a year.	2 io1 volumes, 2,000 GB at 2,000 IOPS each; \$510 a month	2 gp3 volumes, 2,000 GB at 3,000 IOPS each; \$160 a month	Measured peak is 700 IOPS	Modify back to io1
S-1 Dev, partner non-production, and perf-test schedules. Beth Morrow; marketplace-infra PR #1811; May 28. Saves \$24,183 a year.	11 hosts running 730 hours a month; \$2,610 a month	8 hosts at 260 hours and 3 perf-test hosts at 96 hours; \$595 a month	Development and partner non-production traffic is zero outside working hours; the perf-test fleet is used for on-sale rehearsals a few times a month	Remove the schedule tag; the hosts start within minutes
N-2 One NAT gateway per non-production VPC. Beth Morrow; seatlark-landing-zone PR #417; May 28. Saves \$3,154 a year.	12 NAT gateways in 4 non-production VPCs; \$743 a month	4 NAT gateways, one per VPC; \$480 a month	Non-production needs no availability across three zones	Re-add NAT gateways from the landing-zone template
L-2 Debug logging off in two services. Aaron Feld; marketplace-infra PR #1810; May 28. Saves \$3,600 a year.	1,600 GB a month of debug logs ingested; \$848 a month	1,000 GB a month at info level; \$548 a month	1,600 GB a month of debug ingestion across two services	Set the level back
Total	159 resources		\$8,230 a month saved, estimated	\$98,760 a year

The waste register

Waste is the one group where a change can destroy something. The rule for the register is that every deletion has a signed line, and a copy is kept wherever someone could want the thing again. 31 resources went through it this week.

The waste register by finding. The line for each resource follows.

ID	What was removed	Resources	Size	MonthlySigned by	Kept	If it is wanted again
W-1	Unattached gp2 volumes from test cutovers (10 x 500 GB)	10	5,000 GB	\$500Beth Morrow	One snapshot of the volume	Create a volume from the snapshot
W-2	MGN replication servers and staging volumes left from completed waves (14 servers, 5,600 GB gp3)	14	5,600 GB	\$661Beth Morrow	The colo source server, kept until the lease ends	Reinstall the replication agent on the source server
W-3	Two idle Application Load Balancers from on-sale rehearsals	2	0 GB	\$33Aaron Feld	The configuration, saved to the change ticket	Recreate from the marketplace-infra stack
W-4	Snapshot and AMI sprawl: 8 TB orphaned MGN and AMI snapshots deleted, no-expiry lifecycle policies trimmed (5 TB), 4 TB of wave snapshots to the archive tier, 1.8 TB of RDS manual snapshots	5	18,800 GB	\$951Beth Morrow	Nothing: no volume or image references them; Nothing: no instance launches from them; The last 35 days of snapshots; All of it, in the archive tier, for a year; Nothing: the databases keep their automated backups	Not recoverable; the signed line is the record; Not recoverable past 35 days; the policy's schedule is unchanged; Restore from the archive tier on request

All four lines were signed in the change ticket on the day the runbook ran, by the person named: Beth Morrow for the volumes, the MGN servers, and the snapshots, and Aaron Feld for the load balancers. Before any deletion the runbook ran in dry-run mode, and its output, which lists every resource it would touch and the state it expects to find, was attached to the ticket. It then ran for real only on the rows the owner approved, and it skips any resource whose state has changed since the register was written ([The waste runbook](#)).

The signed-off waste register for week 3, one line per resource. "Monthly" is the resource's own list price; for the archived snapshots it is the difference between the standard and archive tiers.

ID	Resource	Account	Size	MonthlySigned by	Ticket	Date	Kept until
W-1	vol-cutover-test-01	marketplace-dev	500 GB	\$50Beth Morrow	CHG-3122	May 27	June 26, 2026
W-1	vol-cutover-test-02	marketplace-dev	500 GB	\$50Beth Morrow	CHG-3122	May 27	June 26, 2026
W-1	vol-cutover-test-03	marketplace-dev	500 GB	\$50Beth Morrow	CHG-3122	May 27	June 26, 2026
W-1	vol-cutover-test-04	marketplace-dev	500 GB	\$50Beth Morrow	CHG-3122	May 27	June 26, 2026

ID	Resource	Account	Size	Monthly	Signed by	Ticket	Date	Kept until
W-1	vol-cutover-test-05	marketplace-dev	500 GB	\$50	Beth Morrow	CHG-3122	May 27	June 26, 2026
W-1	vol-cutover-test-06	marketplace-dev	500 GB	\$50	Beth Morrow	CHG-3122	May 27	June 26, 2026
W-1	vol-cutover-test-07	marketplace-staging	500 GB	\$50	Beth Morrow	CHG-3122	May 27	June 26, 2026
W-1	vol-cutover-test-08	marketplace-staging	500 GB	\$50	Beth Morrow	CHG-3122	May 27	June 26, 2026
W-1	vol-cutover-test-09	marketplace-staging	500 GB	\$50	Beth Morrow	CHG-3122	May 27	June 26, 2026
W-1	vol-cutover-test-10	marketplace-staging	500 GB	\$50	Beth Morrow	CHG-3122	May 27	June 26, 2026
W-2	mgn-repl-01	sl-migration	400 GB	\$47	Beth Morrow	CHG-3123	May 27	June 30, 2027
W-2	mgn-repl-02	sl-migration	400 GB	\$47	Beth Morrow	CHG-3123	May 27	June 30, 2027
W-2	mgn-repl-03	sl-migration	400 GB	\$47	Beth Morrow	CHG-3123	May 27	June 30, 2027
W-2	mgn-repl-04	sl-migration	400 GB	\$47	Beth Morrow	CHG-3123	May 27	June 30, 2027
W-2	mgn-repl-05	sl-migration	400 GB	\$47	Beth Morrow	CHG-3123	May 27	June 30, 2027
W-2	mgn-repl-06	sl-migration	400 GB	\$47	Beth Morrow	CHG-3123	May 27	June 30, 2027
W-2	mgn-repl-07	sl-migration	400 GB	\$47	Beth Morrow	CHG-3123	May 27	June 30, 2027
W-2	mgn-repl-08	sl-migration	400 GB	\$47	Beth Morrow	CHG-3123	May 27	June 30, 2027
W-2	mgn-repl-09	sl-migration	400 GB	\$47	Beth Morrow	CHG-3123	May 27	June 30, 2027
W-2	mgn-repl-10	sl-migration	400 GB	\$47	Beth Morrow	CHG-3123	May 27	June 30, 2027
W-2	mgn-repl-11	sl-migration	400 GB	\$47	Beth Morrow	CHG-3123	May 27	June 30, 2027
W-2	mgn-repl-12	sl-migration	400 GB	\$47	Beth Morrow	CHG-3123	May 27	June 30, 2027
W-2	mgn-repl-13	sl-migration	400 GB	\$47	Beth Morrow	CHG-3123	May 27	June 30, 2027
W-2	mgn-repl-14	sl-migration	400 GB	\$47	Beth Morrow	CHG-3123	May 27	June 30, 2027

ID	Resource	Account	Size	Monthly	Signed by	Ticket	Date	Kept until
W-3	alb-rehearsal-1	marketplace-staging		\$16	Aaron Feld	CHG-3124	May 27	none
W-3	alb-rehearsal-2	marketplace-staging		\$16	Aaron Feld	CHG-3124	May 27	none
W-4	snap-orphaned-mgn-conversion	sl-migration	5,000 GB	\$250	Beth Morrow	CHG-3125	May 28	none
W-4	snap-orphaned-test-cutover-amis	sl-migration	3,000 GB	\$150	Beth Morrow	CHG-3125	May 28	none
W-4	snap-lifecycle-no-expiry	sl-backup	5,000 GB	\$250	Beth Morrow	CHG-3125	May 28	none
W-4	snap-golden-pre-cutover	sl-backup	4,000 GB	\$130	Beth Morrow	CHG-3125	May 28	May 28, 2027
W-4	rds-manual-snapshots	sl-backup	1,800 GB	\$171	Beth Morrow	CHG-3125	May 28	none

Some things cannot be undone

Deleted snapshots and AMIs in W-4 are not recoverable, which is why none was deleted that had a volume or image behind it: the orphaned chains had neither, and the no-expiry policy keeps its last 35 days. Deleted RDS manual snapshots are gone too; the databases keep their automated backups. And a log event older than its new retention period is gone the moment the retention applies (L-1). Each has a signed line, and nothing was removed from a resource an owner had not approved.

How each change was made, and how it is undone

9 of the 14 changes went through pull requests, which is the route we prefer: the owner reviews the diff, the pipeline applies it, and the repository history is the rollback record. The other 5 are resources no repository defines (volumes and servers MGN created, snapshots, and a bulk volume-type change), so we wrote a runbook and Seatlark's operators ran it under 5 change tickets, as your change process requires. Vantalect holds no standing write access: we write the pull request or the script and its dry-run output, and a Seatlark engineer merges or runs it.

Where the week-3 changes were made. All four repositories use AWS CloudFormation, with Ansible for the lifted hosts.

Repository or ticket	ChangesPull requests or tickets
seatlark-landing-zone	5N-1 (#415), L-1 (#413), L-3 (#414), S3-2 (#416), N-2 (#417)
marketplace-infra	4D-1 (#1808), E-2a (#1809), S-1 (#1811), L-2 (#1810)
Scripted runbook, 5 change tickets	5W-1 (CHG-3122), W-2 (CHG-3123), W-3 (CHG-3124), E-1 (CHG-3121), W-4 (CHG-3125)

Every change followed the same rules. The owner answered within 2 business days, and their words are kept in [What was declined, and what could still be reversed](#). Nothing in the production request path changed. A volume is snapshotted and the snapshot kept 30 days; a load balancer's configuration is saved to the ticket; the final pre-cutover snapshots go to the archive tier for a year. Each runbook prints what it would do and runs for real only with `--execute`, on rows the owner approved. The previous value of every setting changed is written to a ledger before the change. Every rollback is in the last column of [Change by change](#); the irreversible ones are named in the warning above: deleted snapshots and AMIs, deleted RDS manual snapshots, and log events past their new retention.

Wave 6 and the bill

On Saturday, May 30, 2026, wave 6 moves partner reporting and the affiliate feed generator to AWS. It adds about \$2,600 a month, on a migration Seatlark planned and priced before this engagement started. It is growth, not waste, and it lands in the same invoice as week 3’s savings.

Why the invoice will not fall by this week’s savings. Wave 6 returns 32% of them.

Run-rate arithmetic, per month	Amount
May run rate, the bill at week one (Business Support+ included)	\$51,660
Less the week-3 changes, estimated at list	-\$8,230
Plus wave 6 from May 30	+\$2,600
If nothing else changed	\$46,029
Change on the bill	-\$5,630

The measurement does not care. Week 6 takes each change’s own resources (the volume IDs, the NAT gateway IDs, the bucket) and compares the 7 days before the change with the 7 days after, leaving the change day out. A new wave’s servers appear in no change’s window, so the \$2,600 can neither hide a saving nor fake one. Business Support+, a percentage of usage, will rise with wave 6 and fall a little with every saving; that effect is not counted in either direction.

The bill will keep rising with the migration: wave 7, in July to August 2026, is planned at \$22,000 a month. Week 5 puts guardrails and a wave runbook in place so the waves still to come land right. What those guardrails save against Seatlark’s own plan is cost avoidance: **not measured, not counted toward the gate, and not a saving on the bill.**

What was declined, and what could still be reversed

Three findings were declined at the readout on May 22, 2026. They count for nothing toward the gate, and the table says what each means for this week.

The three findings declined at the readout. They count for nothing in the gate.

ID	Change	Owner	AnnualizedReason, as the owner gave it	What it means now
N-4	Second Direct Connect port from 10 Gbps to 1 Gbps	Owen Pradhan	\$17,082The orders database cutover in July needs failover at full rate. The ports are revisited at colo exit.	The 10 Gbps ports stay as they are. The Direct Connect prices are now taken from the AWS Price List API, not assumed, so the number holds when the ports are revisited.
R-3	Listing ingestion workers: 4 x m5.4xlarge to m5.2xlarge	Imani Brooks	\$13,455Compute Optimizer never saw an on-sale. Show me a load test first. Rescheduled as a perf-test exercise after the freeze, on the remaining roadmap.	The ingestion workers stay at m5.4xlarge. The perf-test exercise after the freeze is the test Imani Brooks asked for.
S-2	Schedule: staging 6 AM to 10 PM Eastern on weekdays	Aaron Feld	\$5,308Staging runs the overnight on-sale rehearsals and the partner acceptance tests in other time zones; the schedule would interrupt both.	Staging keeps running every hour. The dev and partner non-production schedules in S-1 are unaffected.

Nobody argued with these, and we did not push. Each owner's reason is a fact about their workload that the Analysis did not have: staging runs overnight on-sale rehearsals and partner acceptance tests, the ingestion workers have never been shown an on-sale on a smaller size, and the orders cutover in July needs the second port's full rate. Together they are \$35,846 a year of the Analysis's total.

One approved item is not in this report's numbers, and could still be reversed. S3-3, deleting the regenerable image renditions of past events, is worth \$3,109 a year and was approved at the readout by Diego Saldana. It is the last irreversible deletion in the roadmap (W-4 and L-1 have already removed data that does not come back, and each has a signed line), so it runs last, after partner engineering confirms nothing links to those renditions. We name the risk here because an approved item can still be declined later. Under the engagement's rule, a change approved at the readout and declined afterward counts toward the guarantee at its estimate, and the week-6 report would list it separately.

Honest limits, as of May 29, 2026

- **Estimates, not measurements.** Every figure above is the Analysis's price at list, annualized. The first check against the bill is next week; the measurement that decides the second invoice is week 6, on data through June 15, 2026.
- **Dollars follow the analysis order.** The Analysis prices waste first, then rightsizing, then schedules, storage, databases, network, and logging, so no dollar is counted twice. S-1 is priced on the partner hosts as R-4 will size them, and S3-2 uses the S3 rate calibrated on your organization's combined usage. The monthly column is the difference between the two sides, to the dollar.
- **W-4 can miss.** Incremental snapshots mean the saving depends on which blocks are shared. It is rated Medium for that reason.
- **W-2 and the MGN fee.** The Cost and Usage Report carries no MGN per-server charge for these servers, because none was replicating; W-2 removes the replication servers and staging disks, which is the whole of what billed.
- **E-1 is EBS only.** RDS storage is not a finding: RDS gp3 lists at the same \$0.115 a GB-month as gp2 in us-east-1, so switching saves nothing at these prices.
- **S-1 assumes 96 perf-test hours a month.** Rehearsals decide the real figure, and it can only go up from the plan.
- **S3-2 carries a small one-time charge.** Moving 300,000 objects to Deep Archive costs \$15 in transition requests, which is part of the one-time total above.

- **The CUR finalizes about 48 hours late.** The 7 days after a change made on May 27, 2026 are final on June 5, 2026, and after a change on May 28, 2026, on June 6, 2026.

The exact mechanism

Everything below is the code we used, reduced to what you need to run it again. Nothing changes a resource until it runs against a list its owner approved. Runbook scripts live in `seatlark-landing-zone/finops/runbooks/`, and your operators run them from the shared services account (`sl-shared-services`) with a role assumed in the target account.

The waste runbook

The input is the approved rows of the waste register as a CSV: `id,kind,resource,profile,ticket`. The profile assumes the remediation role in the resource's account. For an MGN row the resource is the source server ID, and for a load balancer its ARN. It runs on the Linux runner (bash, GNU date, jq, AWS CLI v2) and writes one ledger line per action, which stays with the change ticket. Each branch first checks that the resource is still in the state the register recorded, so something started since the Analysis is skipped, not removed. A snapshot that does not complete keeps its volume.

```
BASH
#!/usr/bin/env bash
# waste-remove.sh <approved.csv> [--execute]      (dry run unless --execute)
set -uo pipefail
PLAN="$1"; MODE="dry-run"; [[ "${2:-}" == "--execute" ]] && MODE="execute"
LEDGER="ledger-$(date -u +%Y%m%dT%H%M%S).tsv"
log() { printf '%s\t%s\n' "$(date -u +%FT%TZ)" "$*" | tee -a "$LEDGER" >&2; }

while IFS=, read -r id kind resource profile ticket; do
  [[ "$id" == "id" ]] && continue
  export AWS_PROFILE="$profile" AWS_DEFAULT_REGION=us-east-1
  case "$kind" in
    unattached-volume)
      # W-1
      state=$(aws ec2 describe-volumes --volume-ids "$resource" --query 'Volumes[0].State' --
output text)
      [[ "$state" == "available" ]] || { log "SKIP $id $resource is $state"; continue; }
      [[ "$MODE" == "execute" ]] || { log "DRY $id snapshot, then delete $resource";
continue; }
      snap=$(aws ec2 create-snapshot --volume-id "$resource" --description "$ticket $id" \
--tag-specifications "ResourceType=snapshot,Tags=[{Key=Retain-Until,Value=$(date -u -d
'+30 days' +%F)}]") \
--query SnapshotId --output text) || { log "FAIL $id snapshot"; continue; }
      aws ec2 wait snapshot-completed --snapshot-ids "$snap" || { log "FAIL $id $snap
incomplete; volume kept"; continue; }
      aws ec2 delete-volume --volume-id "$resource" && log "OK $id deleted $resource, kept
$snap" ;;
    mgn-source-server)
      # W-2
```

```

    life=$(aws mgn describe-source-servers --query "items[?
sourceServerID=='$resource'].lifeCycle.state | [0]" --output text)
    [[ "$life" == "CUTOVER" ]] || { log "SKIP $id $resource is $life"; continue; }
    [[ "$MODE" == "execute" ]] || { log "DRY $id disconnect and archive $resource";
continue; }
    aws mgn disconnect-from-service --source-server-id "$resource" >/dev/null \
    && aws mgn mark-as-archived --source-server-id "$resource" >/dev/null && log "OK $id
archived $resource" ;;
    idle-load-balancer) # W-3
        bytes=$(aws cloudwatch get-metric-statistics --namespace AWS/ApplicationELB --metric-name
ProcessedBytes \
        --dimensions Name=LoadBalancer,Value="${resource#*:loadbalancer/}" --statistics Sum --
period 86400 \
        --start-time "$(date -u -d '-90 days' +%FT%TZ)" --end-time "$(date -u +%FT%TZ)" --query
'sum(Datapoints[].Sum)' --output text)
        [[ "$bytes" == "0" || "$bytes" == "0.0" ]] || { log "SKIP $id $resource carried $bytes
bytes"; continue; }
        [[ "$MODE" == "execute" ]] || { log "DRY $id save, then delete $resource"; continue; }
        aws elbv2 describe-load-balancers --load-balancer-arns "$resource" > "$id-lb.json" #
attach to the ticket
        aws elbv2 describe-target-groups --load-balancer-arn "$resource" > "$id-tg.json"
        aws elbv2 describe-listeners --load-balancer-arn "$resource" > "$id-listeners.json"
        aws elbv2 delete-load-balancer --load-balancer-arn "$resource" && log "OK $id deleted
$resource"
        for tg in $(jq -r '.TargetGroups[].TargetGroupArn' "$id-tg.json"); do aws elbv2 delete-
target-group --target-group-arn "$tg" || log "WARN $tg kept"; done ;;
    esac
done < "$PLAN"

```

MGN archives only a source server in the CUTOVER or DISCONNECTED state, so the script disconnects first. Disconnecting stops replication at once, and MGN removes the replication servers and staging disks it created within 90 minutes; launched cutover instances and the colo source server are untouched.

Snapshots, volumes, and the schedule

W-4 has three parts. Deregister an image before deleting its snapshots. Move the pre-cutover final snapshots to the archive tier (a full copy, billed on its full size; the 90-day minimum never applies to a one-year hold; \$0.0125 a GB-month against \$0.05 standard). Set the daily policy's retention to 35 days so the no-expiry chain stops growing.

BASH

```

aws ec2 deregister-image --image-id <ami-id> && aws ec2 delete-snapshot --snapshot-id
<snapshot-id>
aws ec2 modify-snapshot-tier --snapshot-id <snapshot-id> --storage-tier archive
aws dlm get-lifecycle-policy --policy-id <policy-id> > dlm-before.json # the way back
aws dlm update-lifecycle-policy --policy-id <policy-id> --policy-details '{"ResourceTypes":
["VOLUME"],
  "TargetTags": [{"Key": "Backup", "Value": "daily"}], "Schedules": [{"Name": "daily", "CopyTags": true,
  "CreateRule": {"Interval": 24, "IntervalUnit": "HOURS", "Times": ["03:00"]}, "RetainRule":
{"Interval": 35, "IntervalUnit": "DAYS"}}}]'

```

To get an archived snapshot back, `aws ec2 restore-snapshot-tier --snapshot-id <snapshot-id> --temporary-restore-days 7`, which bills the retrieval per GB, then create a volume from it.

gp3 costs \$0.08 a GB-month against \$0.10 for gp2, with 3,000 IOPS and 125 MiB/s included at any size. The change is online with Elastic Volumes. No volume here is larger than 1,000 GB, so gp2's baseline of 3 IOPS a GB never exceeded gp3's included 3,000. The estimate prices every volume at the gp3 defaults; a volume whose measured throughput peak exceeded 125 MiB/s would have been converted with `--throughput` set and priced at the extra rate, and the estimate carries none. The ledger line comes first, so the old values are never lost. A volume can be modified once every 6 hours, so a rollback waits for that.

BASH

```
vol=<volume-id>
read -r type size iops < <(aws ec2 describe-volumes --volume-ids "$vol" --query 'Volumes[0].
[VolumeType,Size,Iops]' --output text)
printf '%s\t%s\t%s\t%s\n' "$vol" "$type" "$size" "$iops" >> volume-ledger.tsv
aws ec2 modify-volume --volume-id "$vol" --volume-type gp3 --iops 3000 --throughput 125
```

E-2a, the staging search volumes, was a pull request instead: `VolumeType: gp3` and `Iops: 3000` (from `io1` at 2,000) on the `AWS::EC2::Volume` in `marketplace-infra`, applied in place.

S-1 is a tag. Instance Scheduler on AWS starts and stops any instance tagged `Schedule` with the name of a schedule in its configuration. The schedule `weekday-12h` runs 12 hours a day Monday to Friday, 07:00 to 19:00 America/New_York (written 07:00 to 18:59 in the scheduler, whose periods end on the last minute); its definition lives in the scheduler stack in `marketplace-infra`, whose syntax differs between the solution's versions, so follow its implementation guide. The perf-test hosts are stopped between rehearsals and started for them, 96 hours a month in the plan. To roll back, remove the tag.

YAML

```
# marketplace-infra/stacks/dev-apps.yaml (excerpt)
DevApp01:
  Type: AWS::EC2::Instance
  Properties:
    InstanceType: m5.xlarge
    Tags: [{Key: Schedule, Value: weekday-12h}]
```

Network and logs

All S3 traffic left through NAT gateways at \$0.045 a GB because no VPC had a gateway endpoint, which costs nothing. N-1 adds one to each of the 6 VPCs in the landing-zone template, and N-2 builds one NAT gateway, not three, outside production. A NAT gateway costs \$0.045 an hour

whether or not it carries traffic. The price of consolidating is availability: a non-production VPC loses outbound access if its one Availability Zone fails. Production keeps three. Reverting the pull request restores the gateways.

YAML

```
# seatlark-landing-zone/templates/vpc.yaml (excerpt)
Conditions:
  ThreeNatGateways: !Equals [!Ref Environment, prod]
Resources:
  S3GatewayEndpoint:
    Type: AWS::EC2::VPCEndpoint
    Properties:
      VpcEndpointType: Gateway
      ServiceName: !Sub com.amazonaws.${AWS::Region}.s3
      VpcId: !Ref Vpc
      RouteTableIds: !Ref PrivateRouteTableIds
  NatGatewayB:
    Type: AWS::EC2::NatGateway
    Condition: ThreeNatGateways          # outside production only NatGatewayA exists
    Properties: {AllocationId: !GetAtt NatEipB.AllocationId, SubnetId: !Ref PublicSubnetB}
  PrivateDefaultRouteB:
    Type: AWS::EC2::Route
    Properties:
      RouteTableId: !Ref PrivateRouteTableB
      DestinationCidrBlock: 0.0.0.0/0
      NatGatewayId: !If [ThreeNatGateways, !Ref NatGatewayB, !Ref NatGatewayA]
```

A log group with no retention keeps every byte forever at \$0.03 a GB-month. L-1 sets 30 days for application and flow-log groups and 90 for audit groups, in the account baseline for new groups and by a one-time loop for the 5 that exist. Logs past the new retention are gone: the loop over `aws logs delete-retention-policy` that the ledger makes possible stops further deletion, but expired events do not come back.

BASH

```
aws logs describe-log-groups --log-group-name-prefix /seatlark \
  --query 'logGroups[?retentionInDays==`null`].logGroupName' --output text | tr '\t' '\n' |
while read -r group; do
  days=30; [[ "$group" == */audit ]] && days=90
  printf '%s\t%s\n' "$group" "$days" >> retention-ledger.tsv
  aws logs put-retention-policy --log-group-name "$group" --retention-in-days "$days"
done
```

L-2 sets `app_log_level: info` (it was debug) in each service's Ansible role defaults, cutting ingestion from 1,600 to 1,000 GB a month at \$0.50 a GB. L-3 delivers VPC flow logs to S3 at \$0.25 a GB, not to CloudWatch Logs at \$0.50, into a bucket with a 30-day lifecycle rule.

The flow-log resource is an `AWS::EC2::FlowLog` with `LogDestinationType: s3` and a `LogDestination` of the bucket's ARN.

The migration backups, and verifying on the bill

S3-2 moves the 28 TB of colo database dumps and VM exports to S3 Glacier Deep Archive, which bills a 180-day minimum at \$0.00099 a GB-month. No expiry is set: Owen Pradhan decides what happens after the first year, and deleting 28 TB of dumps would be its own signed line. A restore takes hours, not milliseconds.

YAML

```
# seatlark-landing-zone/templates/backup-bucket.yaml (excerpt)
LifecycleConfiguration:
  Rules:
    - Id: backups-to-deep-archive
      Status: Enabled
      Transitions: [{StorageClass: DEEP_ARCHIVE, TransitionInDays: 0}]
```

Verify each change the next day against the Cost and Usage Report, by resource ID. The same query on a window of 7 days either side is the week-6 measurement, with the change day left out. A deleted resource has no rows after the change, so read its missing days as zero cost. Seatlark held no commitments before June 3, 2026, so before that date cost is the same on any basis; the query below reads covered usage at its public On-Demand price and everything else at unblended cost, the basis week 6 uses, so a change made after the Savings Plan exists is not credited with the plan's discount. C-1 is measured separately, by the plan's ARN. The export finalizes about 48 hours late, so the next-day check reads its latest finalized day.

SQL

```
SELECT date(line_item_usage_start_date) AS usage_day,
       sum(CASE line_item_line_item_type
             WHEN 'SavingsPlanCoveredUsage' THEN pricing_public_on_demand_cost
             WHEN 'DiscountedUsage' THEN pricing_public_on_demand_cost
             ELSE line_item_unblended_cost END) AS cost
FROM cur2.seatlark_cur
WHERE line_item_resource_id = '<resource-id>'
      AND line_item_usage_start_date >= date_add('day', -7, date '<change-date>')
      AND line_item_usage_start_date < date_add('day', 8, date '<change-date>')
GROUP BY 1 ORDER BY 1;
```

What comes next

Week 4 runs from Monday, June 1, 2026, to Friday, June 5, 2026, and it is the last week production can change before the freeze that runs June 4, 2026 to June 14, 2026. It rightsizes the production API tier and the partner APIs on June 2, 2026, rolling and with an instant rollback, puts the ticket-scan lifecycle rule and the CloudWatch agent change in place on June 1, 2026, and takes the commitment decision with Lydia Brandt on June 3, 2026.

What week 4 plans, ranked by date.

ID	Planned change	Owner	Date	Annualized, estimated
R-4	Non-production: staging search nodes to r5.xlarge, partner non-production to m5.xlarge	Beth Morrow	Jun 1	\$9,461
L-4	CloudWatch agent: per-mount and per-process metrics dropped (2,400 custom metrics to 1,000)	Beth Morrow	Jun 1	\$5,047
S3-1	Ticket-scan uploads to S3 Glacier Instant Retrieval after 120 days (26 TB)	Kara Lindqvist	Jun 1	\$5,674
R-1	Marketplace API tier: 6 of 12 m5.2xlarge to m5.xlarge; the web tier keeps its on-sale headroom	Aaron Feld	Jun 2	\$10,092
R-2	Partner APIs: 6 x m5.2xlarge to m5.xlarge	Diego Saldana	Jun 2	\$10,092
C-1	Compute Savings Plan, one year, No Upfront, \$5.00 an hour	Lydia Brandt	Jun 3	\$15,950

E-2b and D-2 are not in week 4. Both were approved at the readout and deferred (\$20,094 a year between them, counted for nothing toward the gate). Imani Brooks will not touch the search cluster in the week before the biggest on-sale of the summer, so E-2b goes in on Monday, June 22, 2026, after the freeze. D-2 waits on a load test against a restored copy at the new class, which touches no production system and can run during the freeze, and then on the ten business days' notice that Seatlark Connect's contracts require: the notice goes out when the test passes, by June 11, 2026, and the window is Saturday, June 27, 2026.

Week 5 is the freeze: no production change. It builds the guardrails and the wave runbook for the waves still to come. The week-4 report follows on June 5, 2026; its commitment memo records what Lydia Brandt decided on June 3, 2026.

© 2026 Vantalect, LLC. Sample deliverable; Seatlark, Inc. is fictional.

AWS, Amazon Web Services, AWS Graviton, and the AWS service names are trademarks of Amazon.com, Inc. or its affiliates. Ansible is a trademark of Red Hat, Inc. Elasticsearch is a trademark of Elasticsearch B.V. PostgreSQL is a registered trademark of the PostgreSQL Community Association of Canada. MySQL is a trademark of Oracle Corporation and its affiliates. Redis is a trademark of Redis Ltd. Microsoft and SQL Server are trademarks of the Microsoft group of companies. VMware and vCenter are trademarks of Broadcom Inc. NetApp is a trademark of NetApp, Inc. Java is a trademark of Oracle Corporation and its affiliates. Vantalect is not affiliated with or endorsed by any of them. Seatlark, Inc. is fictional.