FIXED-SCOPE • 14 DAYS • $5,000

FinOps Sprint — Sample Report & Detailed Improvement Plan

Expanded report-style sample deliverable: baseline, AWS service deep-dives, implementation log, quantified savings model, and a 14/30/90/180-day roadmap

WHAT THIS IS

A report of work done

Baseline, analysis, changes completed, prioritized improvements, and roadmap.

PRIMARY OUTCOME

Reduce cloud spend

Waste removal, rightsizing, scheduling, and a commitment strategy when justified.

DELIVERY

Implementation included

Not just a report — we implement agreed changes with safety checks.

BEST FIT

$5k–$100k/mo spend

SMBs, agencies, and hosting providers with lean IT/engineering teams.

Prepared by: PlatOps **Prepared for:** Client name (placeholder) **Date:** February 3, 2026

Table of Contents

This report is intentionally detailed and includes example charts, tables, calculations, and timelines.

- 1) Executive Summary
- 2) Baseline & Cost Drivers
- 3) Methodology & Data Sources
- 4) Implementation Log (what we changed)
- 5) Savings Model & Assumptions
- 6) AWS Service Deep-Dives (EC2, EBS, RDS/Aurora, S3, CloudWatch, Network)
- 7) Prioritized Opportunity Backlog
- 8) Roadmap: 14 / 30 / 90 / 180 Days
- 9) Guardrails & Governance
- 10) Optional Monthly FinOps Management
- 11) Scenario Modeling (Safe / Balanced / Riskier)
- Appendix A) Calculation Examples
- Appendix B) Safety & Change Management

How to read this report

- **Baseline:** what you're spending and where it's going.
- **Implementation log:** what we changed during the sprint and how we validated safety.
- **Deep-dives:** service-by-service findings and recommendations.
- **Scenario modeling:** three savings outcomes based on risk tolerance.

This is a sample of a report-style deliverable produced after a FinOps Sprint. It documents what we analyzed, what we implemented within the sprint, what remains, and a realistic plan for the next 14 / 30 / 90 / 180 days. Numbers shown are examples and are tailored to your environment during kickoff.

BASELINE (EXAMPLE)

$12,000 / month current AWS spend

Top drivers: EC2, RDS/Aurora, EBS, and data transfer/NAT.

SPRINT FEE

$5,000 fixed for a 14-day sprint (implementation included)

EXPECTED IMPACT (EXAMPLE)

$2,000–$4,000 / month savings by Day 14

Plus guardrails to reduce cost creep and catch cost spikes early.

PAYBACK (EXAMPLE)

~1–2 months to recover sprint fee

UNALLOCATED SPEND

12%

Target: <5% by Day 30 (owners + tags).

NON-PROD PORTION

28%

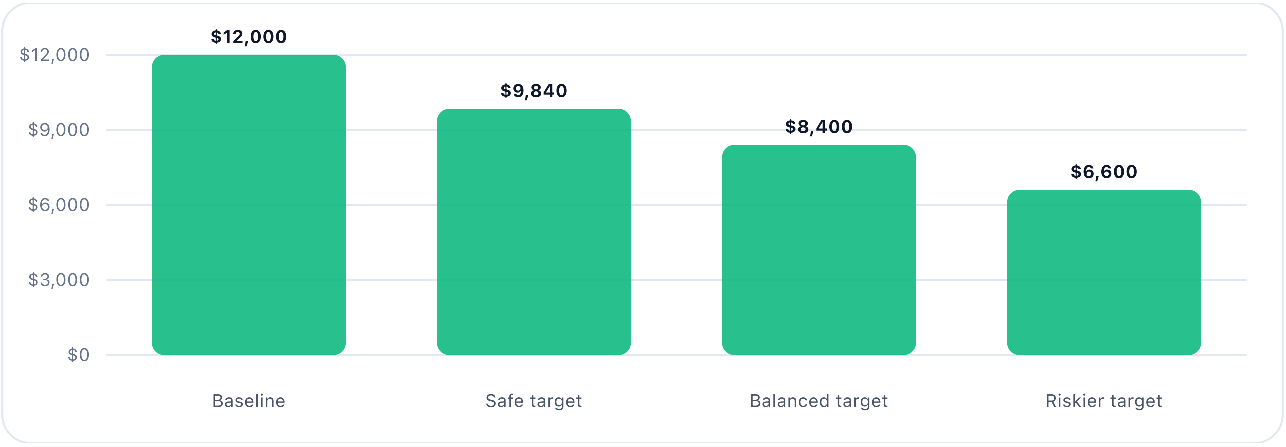
Fastest safe savings via scheduling.

SPRINT GOAL

15–35%

Savings range depends on current maturity + utilization.

Baseline vs target spend (example) — Illustrative (not a guarantee)

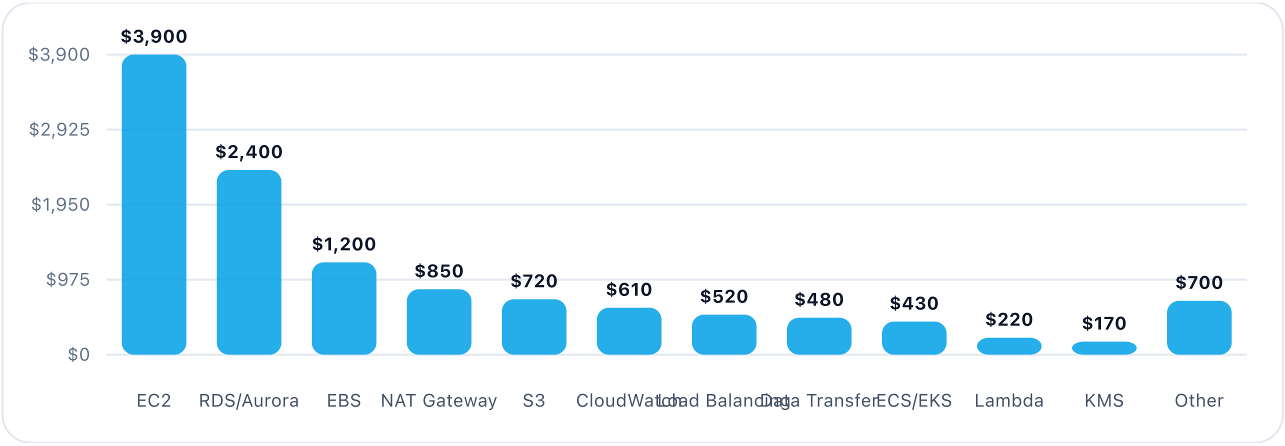


We establish a baseline first so savings can be measured and verified.

Baseline metrics (example)

METRIC	CURRENT (EXAMPLE)	WHY IT MATTERS
Monthly cloud spend	$12,000	Baseline for ROI tracking
Unallocated/unknown spend	12%	Blocks accountability and budgeting
Non-production spend	28%	Typically the fastest safe savings
Commitment coverage	Low	May unlock steady discounts if usage is stable
Budgets/anomaly alerts	Inconsistent	Cost spikes go unnoticed until the invoice arrives

Monthly spend by AWS service (example) — Used to prioritize work by ROI



Top cost line items (example)

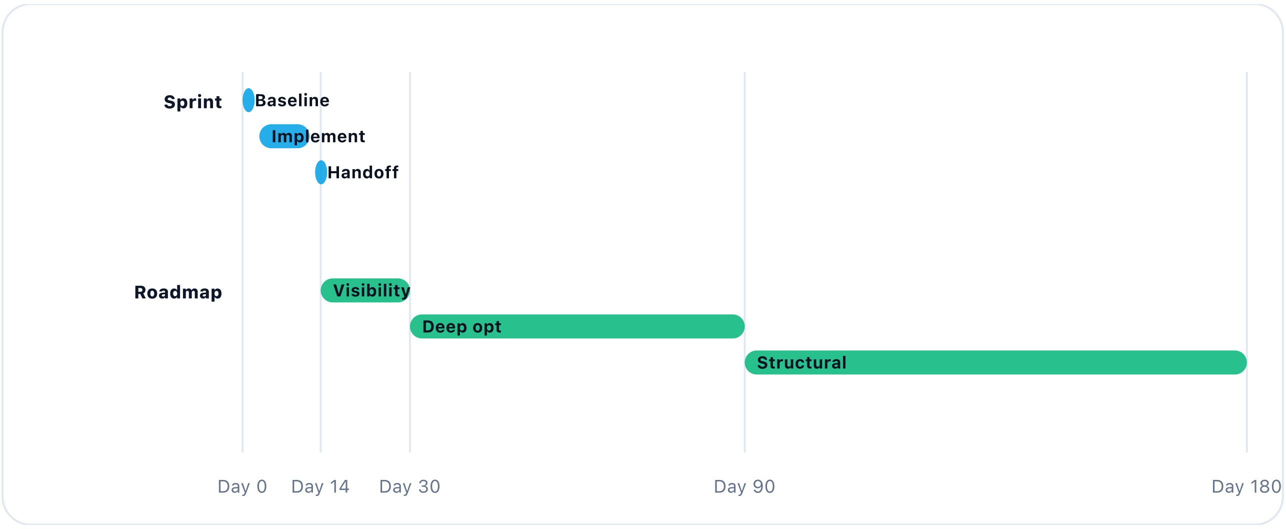
LINE ITEM	OWNER	MONTHLY	NOTES
EC2 compute (prod)	Platform	$2,100	Overprovisioned instances + low CPU utilization
EC2 compute (non-prod)	Engineering	$1,800	Always-on dev/staging; candidates for scheduling
RDS/Aurora instances	Data	$1,600	Sizing + Multi-AZ + storage growth
RDS snapshots/backups	Data	$520	Retention policy too long for non-prod
EBS gp3 volumes	Platform	$880	Orphaned volumes + over-allocated IOPS
NAT Gateway hours	Platform	$520	Multiple NATs + high cross-AZ traffic
NAT Gateway data processing	Platform	$330	Egress + internal traffic through NAT
CloudWatch Logs ingestion	Security	$310	Verbose logging; retention too long
CloudWatch metrics/custom	Platform	$300	High-cardinality metrics
S3 storage	Engineering	$420	Lifecycle missing for non-prod/archives
ALB/NLB	Platform	$520	Idle LBs and extra listeners
Data transfer out	Platform	$480	CDN opportunities + egress hotspots

We combine billing data (what you pay) with utilization/telemetry (what you actually use) to build a prioritized plan that is safe to implement. During a sprint we focus on high-ROI, low-to-medium risk changes first.

Primary sources

- AWS Cost Explorer + detailed service filters
- AWS Cost and Usage Report (CUR) for granularity and attribution
- CloudWatch metrics and logs (utilization, retention, ingestion)
- Resource inventory (instances, volumes, snapshots, EIPs, load balancers)
- Architecture context (prod vs non-prod, SLAs, change windows)

Delivery timeline (example) — How the sprint and roadmap phases stack



How we prioritize

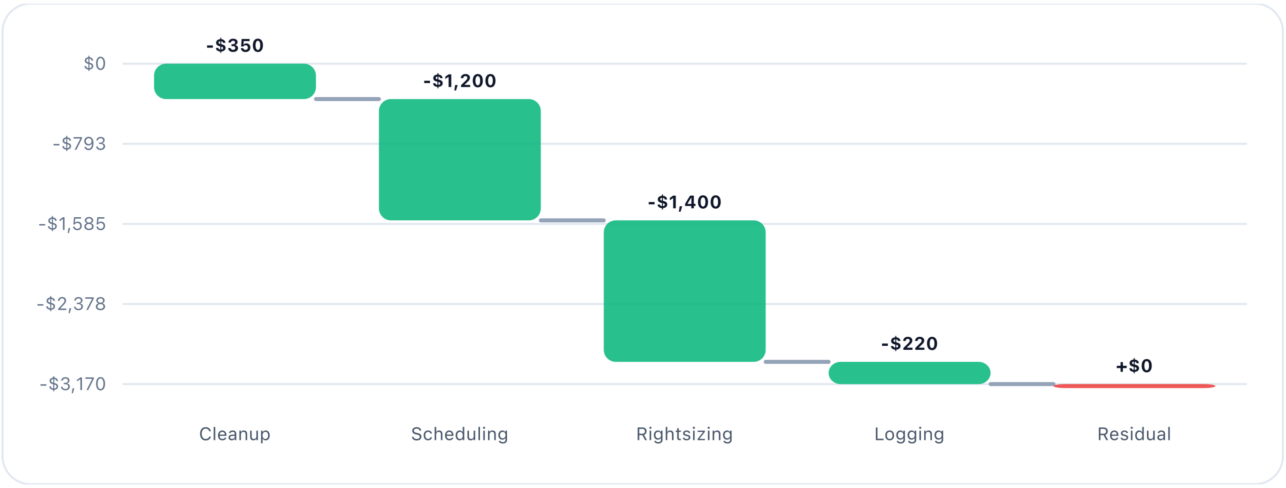
- **ROI:** expected savings range vs effort
- **Safety:** blast radius, rollback options, maintenance windows
- **Durability:** prevents regressions (guardrails, automation, ownership)

This section shows the style of change log included in a real sprint: what we changed, expected impact, and how we validated production safety.

Implemented changes (example)

WORK ITEM	STATUS	EST. MONTHLY SAVINGS	RISK	VALIDATION / SAFETY
Orphan cleanup: unattached EBS + stale snapshots + unused EIPs	Completed	$150–$600	Low	Dependency checks + 7-day hold for snapshots
Non-prod scheduling (nights/weekends) for dev/staging	Completed	$500–$1,800	Low	Exempt CI/nightly jobs; health checks
Right-size top EC2 offenders (prod)	Completed	$700–$1,900	Low–Med	Staged rollout + alarms + rollback
CloudWatch log retention + filter noisy streams	Completed	$120–$420	Low	Retention per env + sampling rules
AWS Budgets + Cost Anomaly Detection + alert routing	Completed	Risk reduction	Risk reduction	Escalation tested

Example: savings waterfall (monthly) — Shows how multiple small wins add up



Savings are modeled conservatively. We only claim savings for changes we implement (or for which a committed change is scheduled), and we separate “theoretical” savings from “realized” savings.

Assumptions used in this sample (example)

ASSUMPTION	VALUE	WHY
Baseline window	Last 30 days	Smooths out weekly noise
Business hours	Mon–Fri	Used for scheduling estimates
Non-prod shutdown	~64% of hours	Nights + weekends
Rightsizing approach	1 step at a time	Reduces risk; validate after each change
Commitments	Only with stability	Avoids lock-in and over-commit

Note: Real reports include links to evidence (CUR queries, Cost Explorer screenshots, inventory exports, and change tickets).

The following pages show the depth of analysis per service: what drives spend, what we checked, what we changed, and what we recommend next.

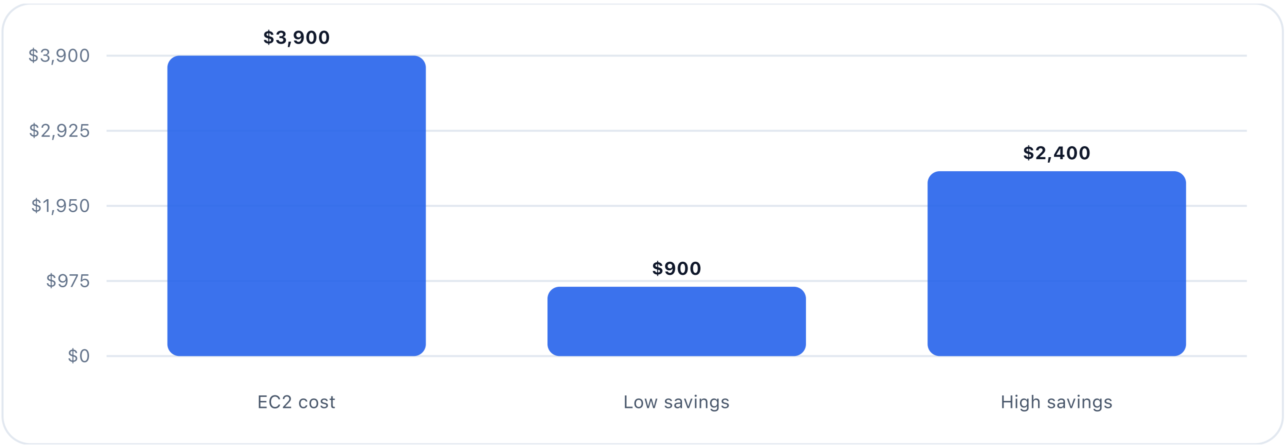
6.1 EC2 (Compute)

EC2 savings typically come from rightsizing, scheduling, and eliminating "always-on" non-prod. We validate utilization patterns and confirm dependencies before changing instance types.

EC2 findings and actions (example)

FINDING	EVIDENCE	ACTION	EST. SAVINGS
Overprovisioned prod instances	Avg CPU < 15%	Downsize one family/size step	$350–$900
Always-on non-prod	No traffic off-hours	Scheduling + auto-start	$500–$1,800
Zombie ASGs	Low utilization	Adjust min/max + scale-in protection	$120–$420

EC2 cost vs potential savings (example) — Conservative savings range



EC2 rightsizing calculation example

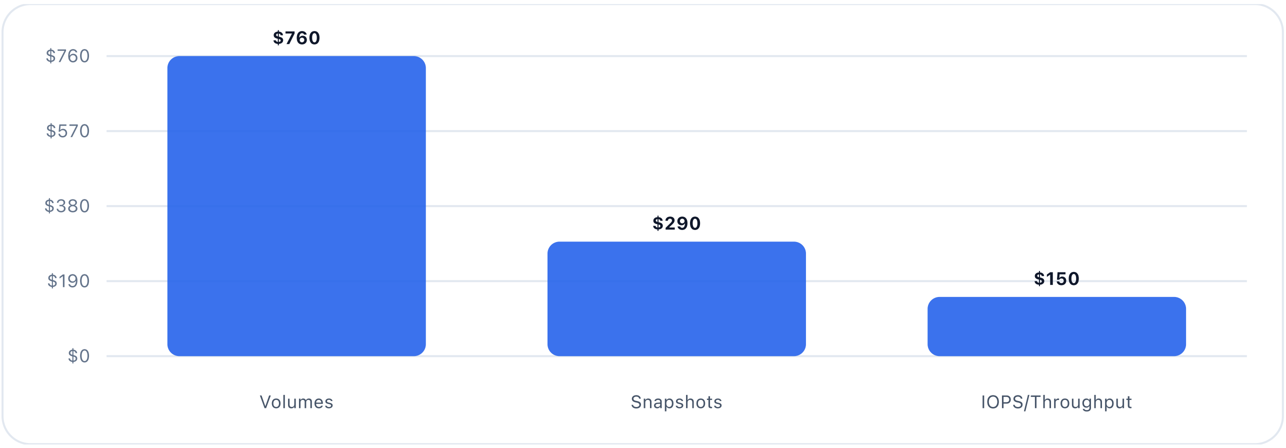
INSTANCE GROUP	CURRENT MONTHLY	PROPOSED MONTHLY	SAVINGS	NOTES
Prod API (3 instances)	$720	$520	$200	One-step downsize + validate
Worker pool (ASG)	$860	$640	$220	Tune min/max and scaling policy
Staging (always-on)	$450	$160	$290	Scheduled downtime

area with careful dependency checks.

EBS calculations (example)

DRIVER	CURRENT	RECOMMENDATION	SAVINGS LOGIC
Unattached gp3 volumes	8 volumes	Delete after verification	Remove full monthly cost
Snapshot retention	90 days	Reduce to 30 days (non-prod)	Reduce stored GB-month
Provisioned IOPS	Over-allocated	Tune IOPS/throughput	Align to observed utilization

EBS cost drivers (example) — Volumes + snapshots + IOPS

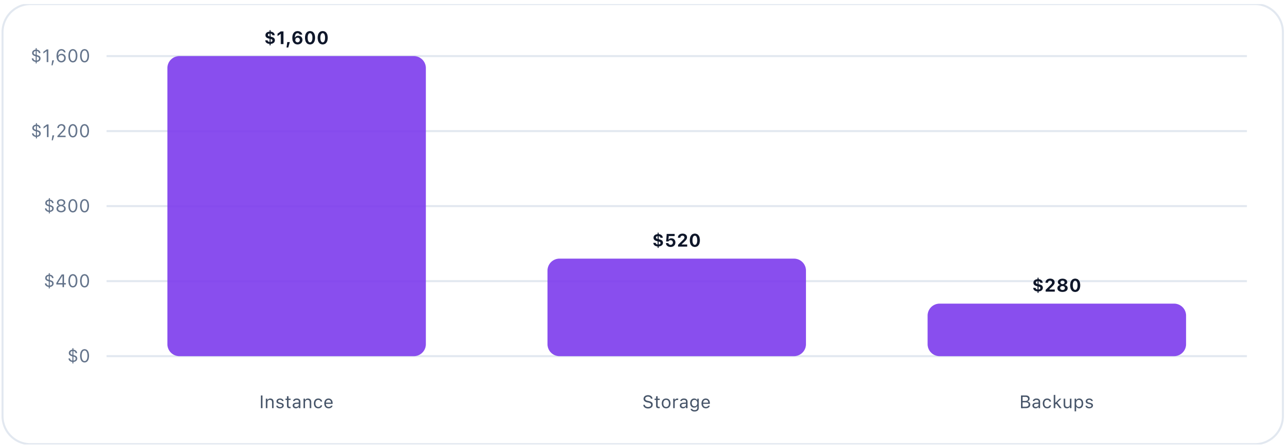


with maintenance windows and validate performance and failover behavior.

RDS/Aurora improvement plan (example)

AREA	CURRENT	ACTION	EST. SAVINGS	RISK
Instance size	db.r6g.large	Downsize after perf test	$180–$520	Med
Backup retention	35 days	Tune per environment	$80–$240	Low
Storage growth	Steady + unmanaged	Set alerts + lifecycle	$50–$160	Low

RDS/Aurora cost drivers (example) — Instance + storage + backups



RDS backup retention savings example

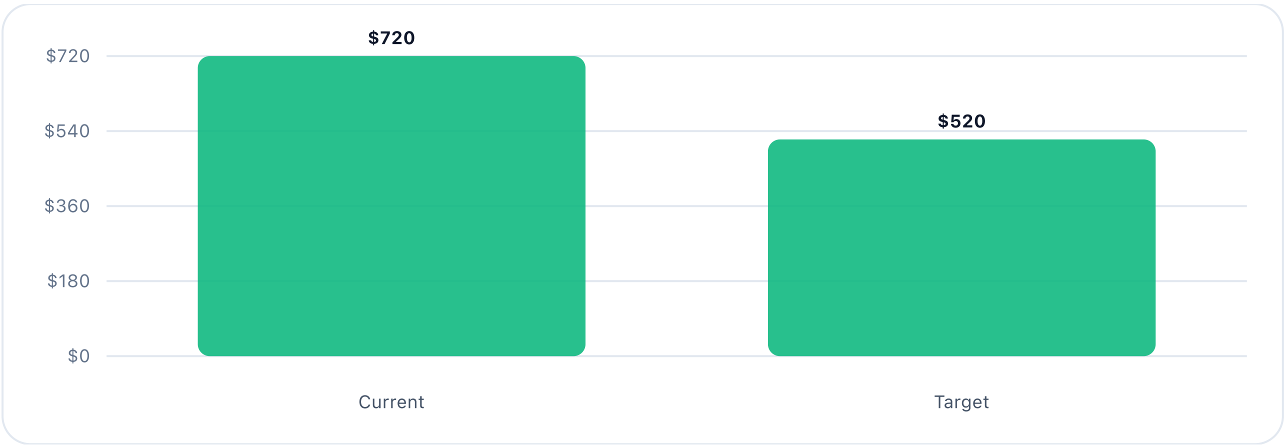
ENV	RETENTION	CHANGE	EST. SAVINGS	RISK
Non-prod	35 days	Reduce to 14–21 days	$40–$120	Low
Prod	35 days	Keep; tune only if agreed	$0–$80	Low–Med

request patterns.

S3 lifecycle example

DATA TYPE	CURRENT	TARGET	EST. SAVINGS
Build artifacts	Standard, no expiry	Expire after 30 days	$40–$120
Logs (non-prod)	Standard, 180 days	Transition to IA + expire	$60–$180
Archives	Standard	Glacier Instant Retrieval	$30–$140

S3 storage optimization (example) — Standard → IA/Glacier + expiration

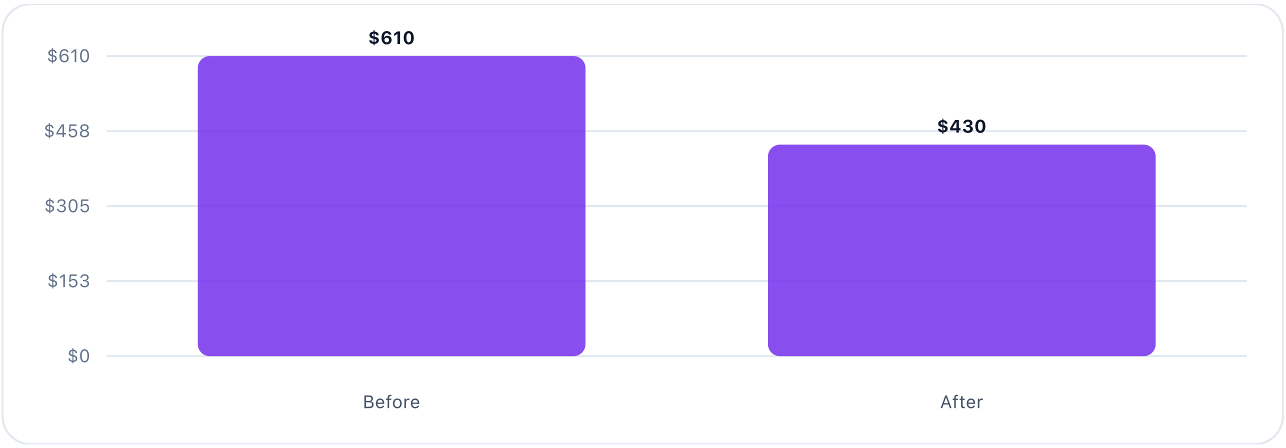


remains actionable.

CloudWatch cost levers (example)

LEVER	ACTION	EST. SAVINGS	NOTES
Retention	Shorten non-prod retention	$60–$180	Keep prod per compliance needs
Ingestion	Drop noisy streams	$40–$160	Add sampling where safe
Custom metrics	Remove high-cardinality metrics	$20–$120	Replace w/ aggregates

CloudWatch: retention impact (example) — Cost drops as GB-month decreases

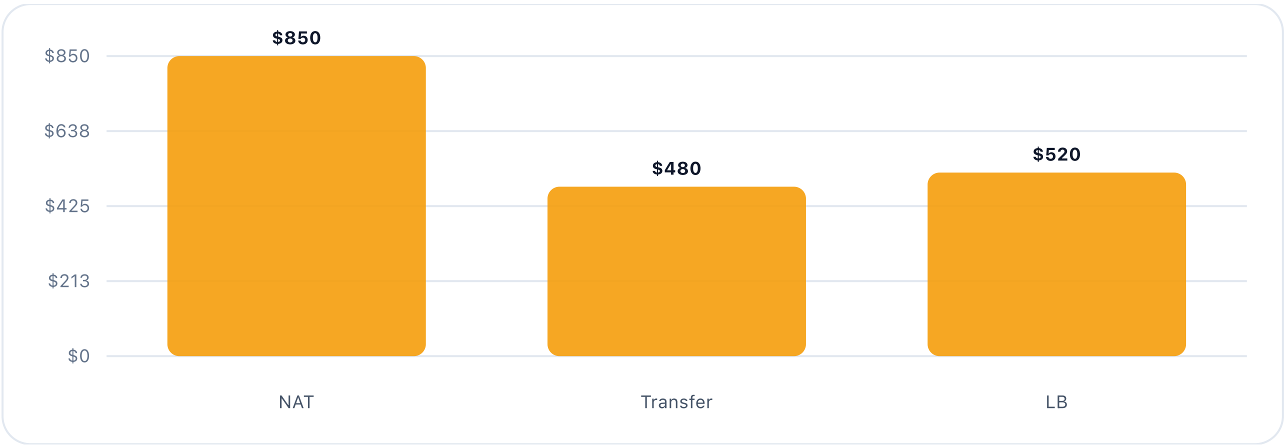


changes, and CDN usage.

Network savings opportunities (example)

AREA	CURRENT	ACTION	EST. SAVINGS
NAT data processing	High	Add VPC endpoints for S3/ECR	\$120–\$420
Cross-AZ traffic	Moderate	Co-locate services / tune subnets	\$60–\$240
Egress	Steady	CloudFront for static + caching	\$80–\$300

Network cost components (example) — NAT + transfer + load balancing



Example NAT Gateway calculation

Savings depend on how much traffic is routed via NAT and which endpoints can be added.

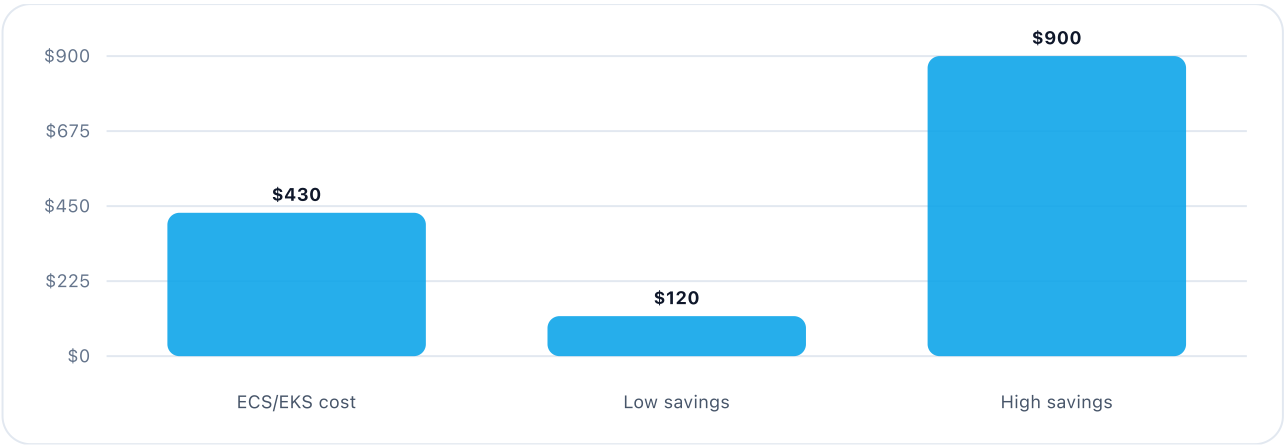
- Assume \$330/mo in NAT data processing is eligible for VPC endpoints.
- Move 50–80% of that traffic off NAT via endpoints → \$165–\$264/mo savings.

binpacking, and “always-on” clusters for dev/test.

ECS/EKS optimization checklist (example)

CHECK	WHAT WE LOOK FOR	ACTION	SAVINGS
Node group utilization	Low CPU/memory	Resize nodes + autoscaler tuning	\$120–\$520
Dev/test clusters	Always-on	Scheduling or scale-to-zero patterns	\$80–\$340
Spot usage	Eligible stateless workloads	Introduce spot with disruption controls	\$100–\$600

Containers: cost vs savings potential (example) — Depends on workload type

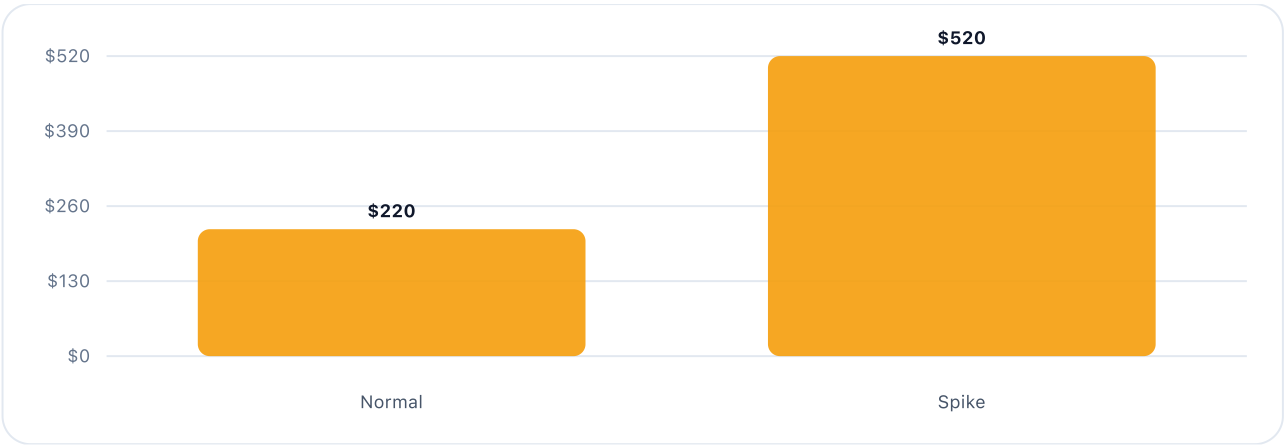


review concurrency, retries, and payload patterns.

Lambda cost controls (example)

DRIVER	SYMPTOM	ACTION	EXPECTED OUTCOME
Retries/timeouts	High invocations	Tune timeouts + idempotency	Lower invocations + fewer failures
Logging	Large log volume	Sampling + reduce verbosity	Lower CloudWatch ingestion
Architecture	Chatty calls	Batching + queueing	Lower duration and transfer

Lambda: typical pattern (example) — Costs often come from spikes

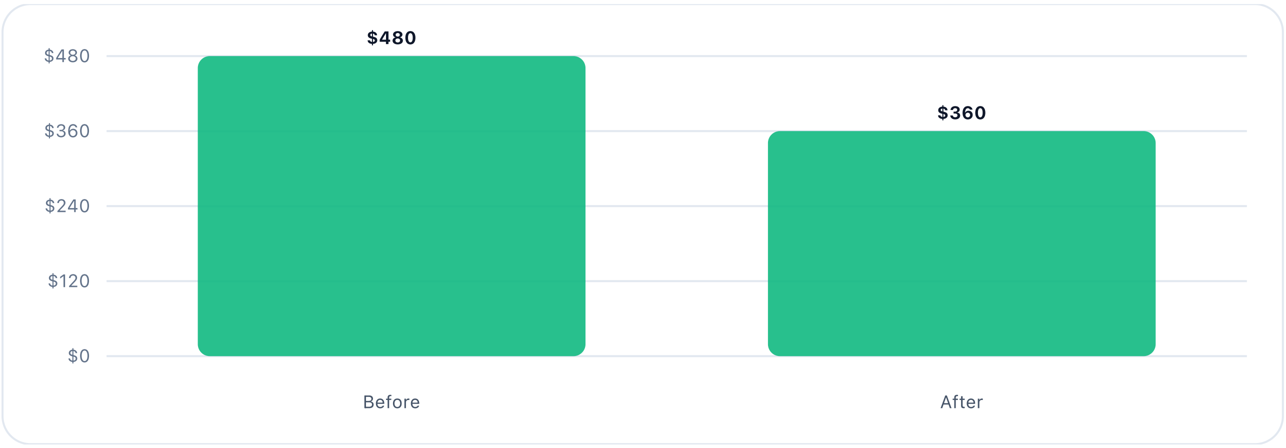


savings approach when egress is non-trivial.

CDN decision table (example)

CONTENT	CURRENT	CDN FIT	SAVINGS PATH
Static assets	Served from origin	High	Cache at edge → lower origin + egress
Public downloads	Direct from S3	High	Cache + reduce repeated egress
API responses	Dynamic	Med	Cache only safe endpoints

Egress savings example (illustrative) — If cache hit rate improves



This backlog captures what remains after the sprint and what to execute next. Priorities are based on ROI, safety, and execution speed.

P0 — Highest priority (do now)

- Fix cost visibility basics: reduce “unknown spend” and assign owners for top cost drivers
- Eliminate recurring waste: enforce cleanup policies for snapshots/volumes/log retention
- Set guardrails: budgets per environment/team + anomaly alerts with escalation rules

P1 — High ROI (do next)

- Database and storage tuning (IOPS, backups, retention, lifecycle policies)
- Rightsizing the next tier of services (containers, autoscaling thresholds, worker pools)
- Data transfer review (cross-zone/region, egress hotspots, CDN/edge configuration)

P2 — Structural improvements (30–90 days)

- Commitment strategy rollout (only when utilization is stable and verified)
- Introduce cost-aware engineering patterns (idle shutdown, autoscale policies, caching)
- FinOps operating cadence: weekly review + monthly executive summary

This roadmap shows what we recommend after Day 14. Each phase has a concrete focus and measurable outcomes.

Post-sprint roadmap (example)

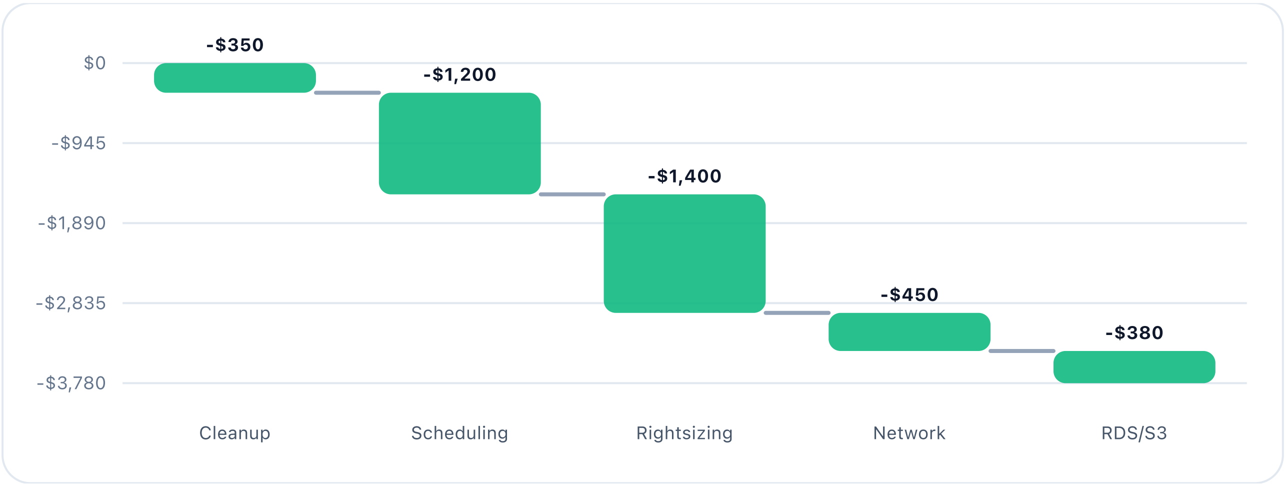
HORIZON	FOCUS	OUTCOMES	OWNER TIME	EXPECTED MONTHLY SAVINGS
Next 14 days	Finish quick wins + stabilize guardrails	• Close remaining orphaned resources • Finalize tagging/ownership for top drivers • Tighten non-prod scheduling policies	~1–2 hrs/week	$2,100–$4,200
30 days	Visibility + repeatable cadence	• Weekly cost review w/ owners • Dashboards + alert tuning • Backlog slice (P1) execution	~2 hrs/week	$2,300–$4,400
90 days	Deep service optimizations	• RDS/Aurora tuning • Network/egress optimizations • Container capacity right-sizing	Engineering time + test windows	$2,600–$5,000
180 days	Structural and commitment strategy	• Savings Plans/RI with approval gates • Cost-aware engineering patterns • FinOps operating model	Steady monthly cadence	$3,000–$5,800

We model ROI from the baseline, using conservative assumptions and approval gates. Below is an example model.

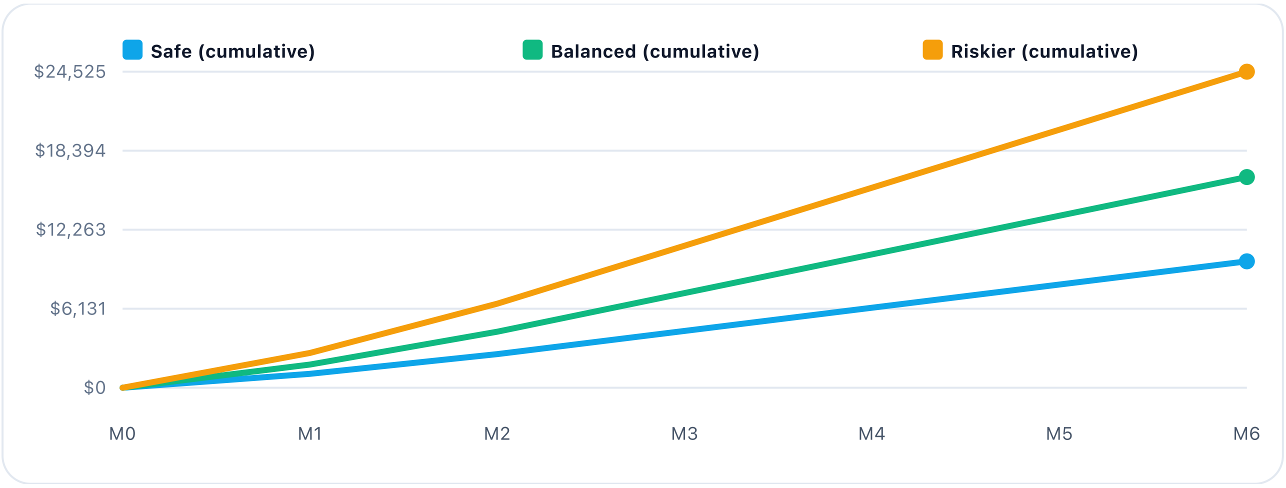
ROI model (example)

TIMEFRAME	BASELINE SPEND	ESTIMATED MONTHLY SAVINGS	FEES	NET BENEFIT
Month 1 (after Day 14)	$12,000	$2,000–$4,000	$5,000 (sprint)	-$3,000 to -$1,000
Month 2	$12,000	$2,300–$4,400	$0	+$2,300 to +$4,400
Month 3	$12,000	$2,600–$5,000	$0	+$2,600 to +$5,000
Month 6 (w/ roadmap)	$12,000	$3,000–$5,800	$0	+$3,000 to +$5,800

Example: path from baseline to target spend — Illustrative, not a guarantee



Cumulative savings over 6 months (three scenarios) — Safe / Balanced / Riskier



Savings are shown as ranges because impact depends on utilization, risk tolerance, and change windows. We prioritize production safety over theoretical savings.

Guardrails ensure savings stick. Governance ensures cost is owned and reviewed like any other operational metric.

Guardrails checklist (example)

CONTROL	WHAT IT DOES	OWNER	CADENCE
Budgets per env/team	Prevents runaway spend	Platform	Weekly
Anomaly alerts	Detects spend spikes fast	Platform	Daily
Tag/owner policy	Reduces unknown spend	Engineering	Weekly
Weekly cost review	Turns findings into action	FinOps lead	Weekly

RACI (example)

ACTIVITY	RESPONSIBLE	ACCOUNTABLE	CONSULTED	INFORMED
Approve medium-risk changes	Platform	Engineering lead	Security	Stakeholders
Rightsize production	Platform	Engineering lead	App owners	Stakeholders
Cost reporting	FinOps lead	CFO/COO	Platform	Leadership
Commitments strategy	FinOps lead	CFO/COO	Platform	Leadership

Measurement plan

- **Weekly:** top 10 drivers, deltas, actions, owners
- **Monthly:** realized savings vs baseline, forecast, commitment coverage
- **Quarterly:** architecture-level optimizations and policy refresh

10) Recommended Next Steps (What we recommend)

Recommended actions after the sprint (typical):

- Weekly 30-minute cost review with owners (top drivers, deltas, and actions)
- Monthly executive summary (savings trend + backlog slice + decisions needed)
- Finish P1 improvements: DB/storage tuning and data transfer hotspots
- If usage is stable: evaluate and phase in commitments with guardrails

Optional: Monthly FinOps Management

If you want ongoing savings and tight control (without hiring a dedicated FinOps role), we can run a lightweight monthly FinOps management cadence.

$1,200–$2,500 / month (typical for ~ $12k/mo AWS spend; depends on account count and complexity)

WHAT'S INCLUDED

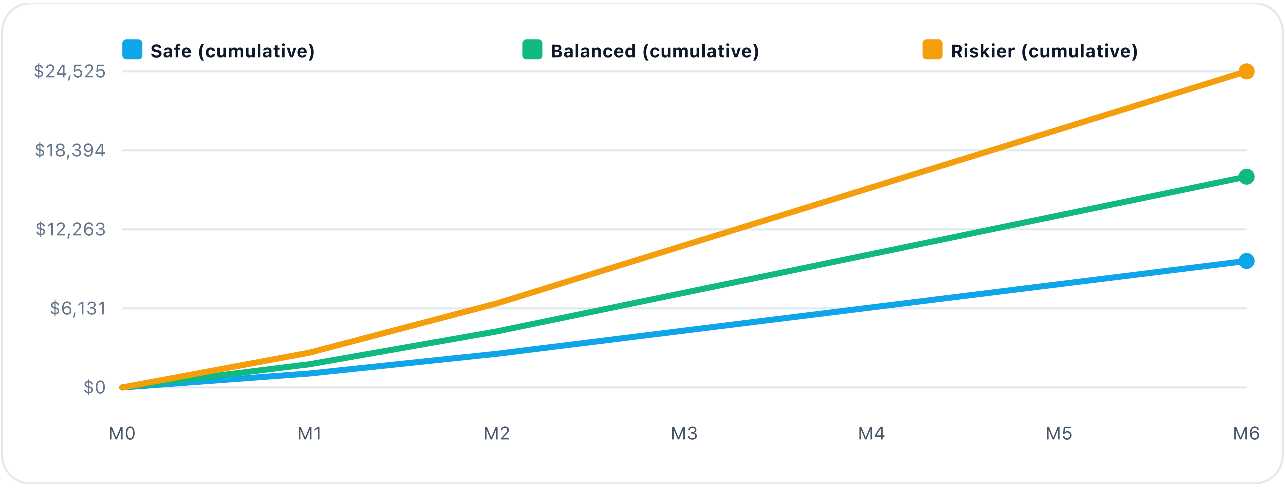
- Weekly cost driver review + action list (owners, due dates, and estimated impact)
- Ongoing guardrail maintenance (Budgets, anomaly alerts, thresholds)
- Commitment management (Savings Plans/Reserved Instances) with approval gates
- Tagging/cost allocation hygiene checks and “unknown spend” reduction
- Monthly executive summary: savings trend, risks, and next backlog slice

Different teams have different risk tolerance. Below are three scenario outcomes showing how savings can vary based on how aggressively you change production capacity and how quickly you adopt structural improvements.

Scenario summary (example)

SCENARIO	DESCRIPTION	ESTIMATED MONTHLY SAVINGS	TARGET MONTHLY SPEND
Safe	Conservative changes first; minimal production risk.	$1,440–$2,160	$9,840–$10,560
Balanced	Typical sprint + roadmap execution; best ROI/safety trade-off.	$2,400–$3,600	$8,400–$9,600
Riskier	Aggressive savings; requires tighter change windows and deeper architecture work.	$3,600–$5,400	$6,600–$8,400

Scenario comparison: cumulative savings — Illustrative 6-month view



Scenario levers (example)

SCENARIO	LEVER	APPROX SHARE OF SAVINGS	CONFIDENCE
Safe	Scheduling (non-prod)	5%	High
Safe	Cleanup (orphans)	2%	High
Safe	CloudWatch retention/noise	1%	Med
Safe	EC2 small rightsizing	4%	Med
Balanced	Scheduling (non-prod)	6%	High
Balanced	Rightsizing (EC2/RDS)	10%	Med
Balanced	Network + NAT optimizations	4%	Med
Balanced	Storage optimization (EBS/S3)	3%	Med
Riskier	Broader rightsizing + capacity refactor	18%	Med
Riskier	Container consolidation (EKS/ECS)	8%	Low–Med
Riskier	Commitments (Savings Plans/RIs)	8%	Low–Med
Riskier	Network topology + egress	5%	Med

OPPORTUNITY	EFFORT	RISK	EST. MONTHLY SAVINGS	NOTES
Right-size overprovisioned compute	Medium	Low–Med	$250–$1,100	Staged rollout + monitoring/rollback plan
Remove orphaned volumes/snapshots/IPs	Low	Low	$100–$500	Safe deletes after dependency verification
Schedule non-prod to shut down nights/weekends	Low	Low	$300–$1,200	Exemptions for CI/nightly workloads
Commitment strategy (Savings Plans/RIs)	Medium	Low	$250–$1,100	Only if utilization is stable and justified

Guardrails (example)

Guardrails we typically implement (tailored to your cloud and org structure):

- Budget thresholds per account/project/team with escalation rules
- Anomaly alerts: spend spikes, unusual resource creation, unexpected data transfer
- A weekly “Top 10 cost drivers” report with owners and actions
- Tagging / allocation recommendations for visibility and accountability

Appendix B) Safety & Change Management

We treat cost optimization like production engineering work: we validate dependencies, roll out in stages, and ensure you can revert changes.

- Approval gates before impactful changes (DB instance size changes, scaling policy changes)
- Verification steps documented per change
- Rollback plan for each medium-risk change
- Change log included in the executive summary

Appendix C) Engagement Inputs (to customize this report)

- Monthly spend range (or last invoice) and primary cloud provider
- Read-only access (preferred) or screen-share implementation sessions
- A single approval path for changes
- Change window rules and rollback requirements

Appendix D) Contact

Book a call: https://platops.com/book-call/

Share a rough monthly spend range and cloud provider, and we’ll confirm fit and outline what we expect to accomplish in 14 days.

Client Name / Title / Date	**PlatOps** Name / Title / Date