

CODELEVEL, AWS cost audit

Sample data

AWS cost audit report

Client	Example company: B2B SaaS, 1 production account and staging, region eu-central-1
Scope	Audit M
Data	Cost Explorer, CloudWatch metrics, June–August 2026
Report date	15 September 2026
Currency	USD, as billed by AWS

1. Summary

USD 4,050 /month Bill today	USD 1,386 /month To recover	34% of the bill That is
---------------------------------------	---------------------------------------	-----------------------------------

I found 10 changes worth USD 1,386 a month in total, or USD 16,632 a year. 8 of them are low risk and bring USD 816 a month; most are configuration changes of a few hours. Only one needs work from your developers.

The largest item is the production database: it has twice the power and memory that three months of metrics call for. The second is staging, which runs around the clock but is only used on working days. I suggest starting with the clean-up that does not touch the application (week 1), and buying the Savings Plan last, only at the new, lower level of usage.

2. Bill today

Average of three full months, net, by service.

Service	USD/month	
EC2 (instances)	1,290	
RDS	1,070	
EBS (volumes and snapshots)	340	
NAT Gateway	300	
CloudWatch	300	
S3	220	
Data transfer out	150	
Load balancers (ELB)	140	
ElastiCache	130	

Service	USD/month
Other (Route 53, KMS, ECR, Secrets Manager)	110 
Total	4,050

3. List of changes

Net amounts per month. The order of implementation is in the plan (section 5).

#	Change	USD/month	Effort	Risk
01	Production RDS: db.r5.xlarge to db.r6g.large (Multi-AZ)	460	1 day	medium risk
02	Staging switched off at night and at weekends	179	4 h	low risk
03	CloudWatch log retention: 30 days instead of forever	136	1 h	low risk
04	VPC endpoints for S3 and ECR instead of traffic through NAT	115	4 h	low risk
05	API on Graviton: 4 × m5.xlarge to 4 × m7g.xlarge	110	2 days	medium risk
06	A 1-year Compute Savings Plan, no upfront payment	158	1 h	low risk
07	Old EBS snapshots: 1.6 TB of 1.9 TB	86	1 h	low risk
08	A lifecycle rule for the S3 log bucket	62	2 h	low risk
09	gp2 volumes to gp3	48	2 h	low risk
10	An unused load balancer and 3 Elastic IP addresses	32	1 h	low risk
Total		1,386		

4. The changes one by one

01 Production RDS: db.r5.xlarge to db.r6g.large (Multi-AZ)

USD 460 a month effort 1 day medium risk no code changes

What I see

The production database averages 14% CPU (41% at peak). ReadIOPS are low and flat, even at peak, so the working set fits in memory. The test on a copy of the database shows whether it fits in 16 GB. It was sized at launch, for traffic that never came.

What to change

A performance test on a copy of the database as db.r6g.large (Graviton, 16 GB), then a class change in a maintenance window. Multi-AZ stays.

Calculation

db.r5.xlarge Multi-AZ about USD 850 a month,
db.r6g.large Multi-AZ about USD 390 a month.

How to undo it

Change the class back, about 15 minutes, with a 1–2 minute Multi-AZ failover.

02 Staging switched off at night and at weekends

USD 179 a month effort 4 h low risk no code changes

What I see

Staging (2 × m5.large and a db.t3.large) runs 168 hours a week. CloudWatch CPU and network metrics show it only works on working days, between 8:00 and 20:00. The rest of the week it sits idle.

What to change

A schedule in EventBridge Scheduler: start at 8:00, stop at 20:00, off at weekends. Manual start with one command.

Calculation

Staging instances USD 280 a month × the 64% of hours when they sit unused.

How to undo it

Switch the schedule off.

03 CloudWatch log retention: 30 days instead of forever

USD 136 a month effort 1 h low risk no code changes

What I see

Every log group is set to “Never expire”. Storage costs USD 170 a month and grows every month.

What to change

Retention of 30 days for application logs and 365 days for audit logs. Older logs, if you need them, exported to S3.

Calculation

About 80% of the stored data is older than 30 days: USD 170 × 0.8.

How to undo it

Retention can be extended at any time, but deleted logs cannot be recovered.

04 VPC endpoints for S3 and ECR instead of traffic through NAT

USD 115 a month effort 4 h low risk no code changes

What I see

The NatGateway-Bytes line is USD 225 a month. Your team ran my query on the VPC Flow Logs: 72% of the data going through the NAT Gateway goes to S3 and ECR, that is to AWS services in the same region.

Calculation

NAT processing $\text{USD } 225 \times 0.72 = \text{USD } 162$, minus about USD 47 a month for the ECR endpoints.

What to change

A gateway endpoint for S3 (free) and interface endpoints for ECR in two zones. Before that I check the bucket policies for conditions on the IP address.

How to undo it

Delete the endpoints and the traffic goes back through NAT.

05 API on Graviton: 4 × m5.xlarge to 4 × m7g.xlarge

USD 110 a month effort 2 days medium risk needs developer work

What I see

The API instances average 9% CPU but use 60% of their memory, so they cannot be made smaller. The processor architecture can change.

Calculation

4 × m5.xlarge about USD 670 a month, 4 × m7g.xlarge about USD 560 a month.

What to change

Build the images for arm64 in CI, test on staging, then replace the nodes one at a time.

How to undo it

Go back to the previous launch template, with no downtime.

06 A 1-year Compute Savings Plan, no upfront payment

USD 158 a month effort 1 h low risk no code changes

What I see

The steady part of production (the API and 3 workers) has run around the clock for over a year, all at on-demand prices.

Calculation

Production after change 05: about USD 990 a month × 0.8 × a discount of about 20%.

What to change

Bought after change 05, for about 80% of the steady usage, so you do not pay for headroom.

How to undo it

A 12-month commitment that cannot be cancelled. You keep paying for it even if you leave AWS. That is why it covers only 80% of usage.

07 Old EBS snapshots: 1.6 TB of 1.9 TB

USD 86 a month effort 1 h low risk no code changes

What I see

Manual snapshots older than 90 days and snapshots of volumes that no longer exist, together 1.6 TB of billed storage (from resource-level Cost Explorer data, not from the volume sizes). None belongs to an AWS Backup plan.

Calculation

1,600 GB × about USD 0.054 per GB a month.

What to change

Deleted once you confirm the list, then a rule in Data Lifecycle Manager.

How to undo it

None: a deleted snapshot is gone. That is why the list goes to you for approval.

08 A lifecycle rule for the S3 log bucket

USD 62 a month effort 2 h low risk no code changes

What I see

3.2 TB of logs older than 90 days in the Standard class, read once every few months. The average object is about 2 MB (from the S3 metrics), so the 128 KB minimum object size does not add to the cost.

Calculation

3,200 GB × a difference of about USD 0.0195 per GB a month.

What to change

A lifecycle rule: after 90 days, move to Glacier Instant Retrieval.

How to undo it

Delete the rule; reading old files costs more, but is still instant.

09 gp2 volumes to gp3

USD 48 a month effort 2 h low risk no code changes

What I see

2 TB of volumes in the older gp2 class. None needs more than 3,000 IOPS and 125 MB/s at peak, which is what gp3 includes in its price.

Calculation

2,000 GB × a difference of about USD 0.024 per GB a month.

What to change

Change the class to gp3 on the fly, with no instance restart.

How to undo it

Change the class back, also on the fly (6 hours after the previous change).

10 An unused load balancer and 3 Elastic IP addresses

USD 32 a month effort 1 h low risk no code changes

What I see

A load balancer left over from an old test environment, with no traffic for 4 months, and 3 IP addresses not attached to any machine.

Calculation

Load balancer about USD 21 a month, addresses about USD 11 a month.

What to change

Deleted once we confirm nothing uses them.

How to undo it

A new load balancer takes a few minutes; the IP addresses will be different.

5. Implementation plan

Week 1	Clean-up with no effect on the application 03 CloudWatch log retention: 30 days instead of forever; 04 VPC endpoints for S3 and ECR instead of traffic through NAT; 07 Old EBS snapshots: 1.6 TB of 1.9 TB; 08 A lifecycle rule for the S3 log bucket; 09 gp2 volumes to gp3; 10 An unused load balancer and 3 Elastic IP addresses.	+USD 479
Week 2	Staging schedule 02 Staging switched off at night and at weekends.	+USD 179
Weeks 3–4	Changes with a test: the database and Graviton 01 Production RDS: db.r5.xlarge to db.r6g.large (Multi-AZ); 05 API on Graviton: 4 × m5.xlarge to 4 × m7g.xlarge.	+USD 570
After 30 days of stable running	Commitment at the new level of usage 06 A 1-year Compute Savings Plan, no upfront payment.	+USD 158

6. What I do not change

- RDS after change 01: a one-year reservation makes sense after 30 days on db.r6g.large. I leave it out of the total, because it depends on the result of the performance test.
- ElastiCache (USD 130 a month): the size fits the load. A reservation only makes sense after a year of stable running.
- Data transfer out (USD 150 a month): a CDN in front of the API would lower it, but needs changes to the application. I describe it separately, without an amount in the total.
- Workers (3 × c5.xlarge): a candidate for Graviton in a second step, once the native libraries are checked.

7. How I calculate

Every amount starts from the average monthly cost of the service over three full months in Cost Explorer, priced at the public AWS price list on the date of the report, in US dollars as AWS bills them. The annual saving is the monthly amount times 12. Changes that affect each other (for example a Savings Plan after the move to Graviton) are calculated in order, so nothing is counted twice.

You do not have to implement anything. If you prefer, I implement the changes myself, in Terraform, in your repository.

Sample data. The company, the account and the amounts do not come from any client. The layout and level of detail are the same as in a real report. The result of the example is not a forecast of your audit. The savings depend on your account.

