

The AWS Cost Audit Checklist

45 checks that usually cut an AWS bill by 20–40% — in the order that saves the most, fastest

By TERRAX — DevOps / cloud engineer running 170+ AWS accounts and 500+ Kubernetes clusters. Free to use and share. Want it done for you? → www.terrax-cloud.com (fixed-price AWS Cost Audit, delivered in 5 business days).

How to use this checklist

1. Open **Cost Explorer** → **Group by: Service**, last 3 full months. Write down your top 5 services — they are 80% of the bill.
2. Go through the sections below **only for those services first**.
3. Every check has: **what to look at** → **how to check (console or CLI)** → **typical saving**. Tick it, note the CAD/month, move on.
4. Do the "quick wins" this week; plan the "projects" with your team.

All commands are **read-only**. Run them with a read-only role (e.g. `ReadOnlyAccess` + `AWSBillingReadOnlyAccess`).

0. Visibility first (30 minutes, saves nothing directly — but everything else depends on it)

- ☐ **Cost allocation tags activated** (Billing → Cost allocation tags): `team`, `env`, `app`. Without them you can't tell who spends what.

- ☐ **AWS Budgets** with alerts at 80% / 100% for each account (free for the first 2 budgets per account).
- ☐ **Cost Anomaly Detection** monitor on all services (free). Catches the forgotten GPU instance in days, not months.
- ☐ **Compute Optimizer** opted in for the whole organization (free; needs 14 days of data — turn it on today).
- ☐ **Multi-account?** Use the payer account's Cost Explorer grouped by *Linked account* — find the 3 accounts that grew most in 90 days.

1. Quick wins — idle and forgotten resources (typical: 5–15% of the bill)

- ☐ **Unattached EBS volumes** — `aws ec2 describe-volumes --filters Name=status,Values=available --query 'Volumes[].[VolumeId,Size,VolumeType,CreateTime]'` → snapshot if needed, then delete. ~CAD 0.11/GB-month (gp3).
- ☐ **Old EBS snapshots** (> 90 days, no AMI uses them) — `aws ec2 describe-snapshots --owner-ids self` sorted by `StartTime`. Use Data Lifecycle Manager for retention.
- ☐ **Unused Elastic IPs** — every public IPv4 is now billed (~CAD 5/month each, attached or not). `aws ec2 describe-addresses` → release the ones with no `AssociationId`.
- ☐ **Idle load balancers** — ALB/NLB with no healthy targets or ~0 requests in CloudWatch (`RequestCount`, 14 days). ~CAD 25–30/month each + LCU.
- ☐ **Stopped EC2 instances** kept "just in case" — you still pay for their EBS. Create an AMI and terminate.
- ☐ **NAT Gateways in dev/test** — each costs ~CAD 45/month + CAD 0.06/GB. One per VPC (not per AZ) in non-prod; turn off overnight where possible.
- ☐ **Forgotten environments** — sort resources by tag `env` / by creation date; demo, PoC and "temp" stacks older than 60 days.
- ☐ **CloudWatch Logs with "Never expire"** — `aws logs describe-log-groups --query 'logGroups[?!retentionInDays].[logGroupName,storedBytes]'` → set 14–90 days.
- ☐ **Idle RDS instances** — `DatabaseConnections` ≈ 0 for 7+ days → snapshot and delete, or stop (auto-restarts after 7 days!).

- ☐ **Unused Kubernetes clusters** — an EKS control plane is ~CAD 100/month before any node. Count clusters with no workloads.
-

2. Right-sizing compute (typical: 10–25% of compute)

- ☐ **EC2 average CPU < 20% and memory < 40%** over 14 days → one size down. Compute Optimizer lists them with the saving.
 - ☐ **Move to current generations:** m5 → m7i/m7g, t2 → t3/t4g, c5 → c7g. Same price or cheaper, 10–40% faster.
 - ☐ **Graviton (ARM)** for anything that runs on Linux containers, Java, Python, Node, Go: ~20% cheaper for the same work.
 - ☐ **Burstable (t3/t4g) for small, spiky workloads;** check `CPUCreditBalance` — "unlimited" mode can cost more than a bigger instance.
 - ☐ **Auto Scaling minimums** set for peak traffic 24/7? Lower the minimum, use target tracking on CPU / requests.
 - ☐ **Dev/test schedules** — stop non-prod EC2/RDS nights and weekends (Instance Scheduler): ~**65% off** those resources.
-

3. Kubernetes / EKS (typical: 20–50% of cluster cost)

- ☐ **Requests vs. real usage** — `kubectl top pods -A` vs. requested CPU/memory. Over-requested pods waste whole nodes.
- ☐ **Karpenter (or Cluster Autoscaler) with consolidation** — removes half-empty nodes automatically.
- ☐ **Spot for stateless workloads** — 60–90% cheaper; mix instance families; keep system and stateful pods on on-demand.
- ☐ **Graviton node groups** for multi-arch images.
- ☐ **One cluster per team/app "because"** → consolidate non-prod clusters (each control plane ~CAD 100/month + add-ons).
- ☐ **EKS extended support** — clusters on old Kubernetes versions cost **6× more per hour** for the control plane. Upgrade.
- ☐ **Unused PersistentVolumes** (`kubectl get pv` with status Released) still bill their EBS.

4. Commitments — only AFTER right-sizing (typical: 20–40% on steady compute)

- ☐ **Compute Savings Plans** for the steady baseline (look at the lowest hourly usage of the last 30 days, not the average).
- ☐ **RDS / ElastiCache / OpenSearch Reserved Instances** for databases that run 24/7.
- ☐ **Coverage and utilization reports** (Cost Explorer) — utilization under 95% means you over-bought.

5. Storage — S3 (typical: 20–60% of S3)

- ☐ **S3 Storage Lens** (free tier) — find the 10 largest buckets.
- ☐ **Lifecycle rules**: Standard → Intelligent-Tiering (or IA after 30 days) → Glacier for archives.
- ☐ **Incomplete multipart uploads** — invisible but billed. Lifecycle rule "abort incomplete multipart uploads after 7 days" on every bucket.
- ☐ **Versioned buckets** with no rule for old versions — add "expire noncurrent versions after N days".
- ☐ **Request costs** (millions of small PUT/GET) — batch, cache, or put CloudFront in front.

6. Data transfer (the "hidden" line — typical: 5–20% of the bill)

- ☐ **NAT Gateway data processing** — S3 and DynamoDB traffic through NAT? Add **free Gateway VPC endpoints**.
- ☐ **Cross-AZ traffic** (`DataTransfer-Regional-Bytes`) — chatty services in different AZs; keep them topology-aware.
- ☐ **Internet egress** — CloudFront is cheaper than direct EC2/S3 egress for public content, and the first 1 TB/month is free.
- ☐ **Interface endpoints (PrivateLink)** you created and no longer use — ~CAD 10/month per AZ each.

7. Databases

- ☐ **Aurora I/O-Optimized vs Standard** — if I/O is > 25% of the Aurora bill, switch to I/O-Optimized.
 - ☐ **gp2 → gp3** for RDS and EBS: ~20% cheaper, same or better performance.
 - ☐ **Multi-AZ on dev/test databases** — rarely needed; halves the instance cost.
 - ☐ **DynamoDB on-demand vs provisioned** — steady traffic is cheaper provisioned with auto scaling; spiky traffic on-demand.
-

8. Licences and extras

- ☐ **Windows / SQL Server licence-included instances** — BYOL or Linux where possible.
 - ☐ **AWS support plan** level vs. what you use.
 - ☐ **Marketplace subscriptions** nobody uses (Billing → Bills → Marketplace).
 - ☐ **CloudTrail**: more than one organization trail writing management events = paying twice.
-
-

Your results

#	Finding	Account / region	CAD / month	Effort (S/M/L)	Done
1					
2					
3					

Total found: CAD ____ / month = CAD ____ / year

Want this done for you?

AWS Cost Audit — CAD 1,500, fixed price, 5 business days, no meetings needed. You give a read-only role; you get a report with every saving ranked by CAD/month, the exact steps, and the risks found on the way. **If I don't find at least CAD 1,500 a year in savings, you don't pay.**
→ terraxcloudhq@gmail.com · www.terrax-cloud.com