

The Databricks Cost-Optimisation Checklist

20 places your Databricks bill leaks, and the action that stops each one.

From Cosmos Thrace, a Databricks Silver Partner. We saved clients \$50M+ in 2025.

HOW TO USE THIS

Work top to bottom. The list is ordered by where the money usually is: compute first, guardrails last. Tick what you've already got covered. Anything left unticked is a candidate for savings. You do not need all of it to see a result. The top five levers alone move most bills.

1 Compute and clusters (the biggest lever)

CHECK	ACTION IF UNTICKED	IMPACT
☐ Clusters auto-terminate when idle	Set aggressive auto-termination on every all-purpose cluster	HIGH
☐ Autoscaling on, with sane min/max	Right-size, then let autoscaling handle the peaks	HIGH
☐ Scheduled jobs run on job clusters, not interactive ones	Move jobs off all-purpose clusters to job clusters (lower DBU rate)	HIGH
☐ Spot / fleet instances for fault-tolerant work	Use spot for workers, keep the driver on-demand	MED
☐ Newest-generation instance types where supported	Switch to newer, cheaper-per-performance families	MED
☐ Cluster policies cap size, auto-term, instance type	Set policies so waste is capped by default, not by memory	MED
☐ Photon tested on your heavy SQL and ETL	Trial Photon where it cuts runtime enough to lower total cost	MED

2 Jobs and workflows

CHECK	ACTION IF UNTICKED	IMPACT
☐ You know your top 10 jobs by cost	Profile them. One job often costs 3x its peers for similar work	HIGH
☐ No jobs silently retrying and burning compute	Alert on repeated failures and long-running outliers	MED
☐ No duplicate pipelines processing the same data twice	Consolidate overlapping jobs	MED
☐ Schedules are staggered, not all firing at once	Spread the load to avoid a spike then idle	MED



3 SQL warehouses

CHECK	ACTION IF UNTICKED	IMPACT
☐ Auto-stop set to the shortest idle that fits	Turn warehouses off when the dashboards aren't being read	HIGH
☐ Warehouse size matched to the workload	Right-size. Bigger is not free	MED
☐ Serverless evaluated for spiky BI	Serverless can cut the idle cost of on-and-off BI	MED

4 Storage and Delta

CHECK	ACTION IF UNTICKED	IMPACT
☐ Small-file problem under control (OPTIMIZE / clustering)	Compact files so queries scan less	MED
☐ Queries cluster/partition on the columns you filter on	Cut the data each query has to read	MED
☐ VACUUM runs regularly within retention	Stop paying to store files nothing reads	LOW
☐ Caching used instead of recomputing	Reuse results where the data hasn't changed	LOW

5 Guardrails (this is what makes savings stick)

CHECK	ACTION IF UNTICKED	IMPACT
☐ Every cluster, job and warehouse is tagged to a team or cost centre	Enforce tags via policy so spend is attributable	VISIBILITY
☐ Budgets and spend alerts are set	Get told before finance does	VISIBILITY
☐ Usage and billing are on a dashboard someone looks at	Build a cost view off the system tables	VISIBILITY
☐ One named person owns FinOps	Savings creep back without an owner	DURABILITY

The one thing most teams miss

Finding the savings is the easy half. Keeping them is the hard half. Every bill we've taken down crept back within a quarter when there were no guardrails. Section 5 is not optional. It is what turns a one-off cleanup into a lower bill that stays lower.

Want us to run this for you?

You can work this list yourself. Many teams do, and it helps. If you'd rather have it found and implemented without pulling your engineers off delivery, that's exactly what our Cost-Optimisation Accelerator does.

→ We start with a **free cost assessment**: we tell you, in euros, what we think we can save.

→ If we can't make the case, you owe nothing.

Book a free cost assessment →

cosmosthrace.com/databricks-cost-optimisation-accelerator

