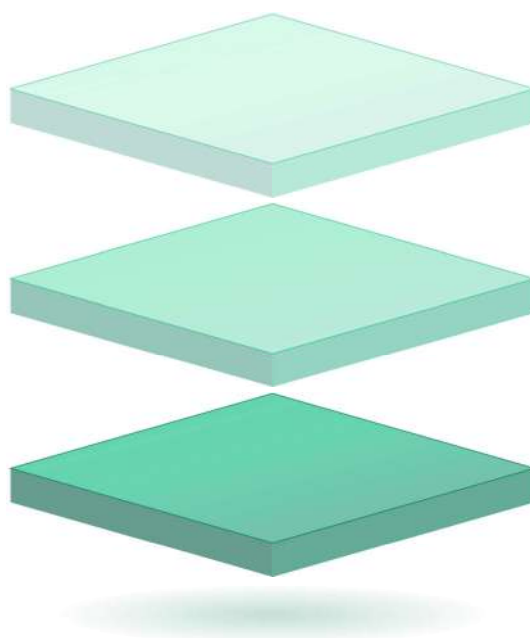


Databricks, optimized at **all layers.**

Continuous optimization that runs inside your Databricks workspace, finds the waste, and ranks every fix by impact.



17+

OPTIMIZATION
ENGINES

20–35%

LOWER INFRA
COST

Daily

CONTINUOUS
ANALYSIS

Zero

DATA LEAVES
YOUR WORKSPACE

01 THE PROBLEM

You already own tools. Each one stops short.

Every team runs a stack to control Databricks cost. Each one answers part of the question, then hands the rest back to an engineer. Lakemine picks up where they stop.

WHAT TEAMS RUN TODAY	WHERE IT STOPS	WHAT LAKEMINE DELIVERS
Databricks system tables & dashboards	Report what happened. An engineer still has to find the cause and decide what to change.	Diagnoses the cause automatically and returns a ranked fix with an estimated saving .
Cloud cost and FinOps platforms	Show spend by account and workspace, without the workload context that explains it.	Ties every dollar to the cluster, job, and query that caused it, then exports it back to Apptio.
Infrastructure monitoring and APM	Hosts and CPU look healthy while spill, skew, idle clusters, and runtime drift stay invisible.	Detects spill, idle compute, runtime drift, and autoscaling faults at the Databricks layer.
Home-grown scripts and notebooks	Accurate the day they are written, then stale as runtimes and workloads change.	Runs every day against the live environment, so findings never go stale.
Consulting engagements	A point-in-time tune-up. The environment drifts back within a quarter.	Holds the gains continuously , catching drift as new teams and workloads arrive.

STAGE 01

Detect

STAGE 02

Diagnose

STAGE 03

Prioritize

STAGE 04

Apply

Existing tooling covers the first two stages. The last two, the ones that actually change the bill, are where Lakemine operates.

02 WHY IT PERSISTS

The environment moves faster **than the tuning.**

Four forces push Databricks spend up, and they compound as the platform grows.

DRIFT

Workloads change weekly

New teams, AI workloads, and warehouses arrive constantly. A cluster sized right last quarter is oversized today.

SCALE

Too wide to tune by hand

Warehouses, clusters, pipelines, serverless, Photon, autoscaling, and spot policy each fail differently.

SKILL

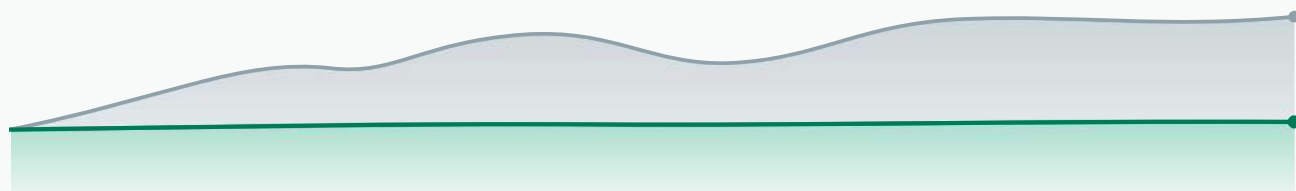
Spark expertise is scarce

The few who know Spark internals are booked on delivery. Tuning waits until the bill forces it.

COST

Waste compounds quietly

Idle compute, oversized nodes, and spill rarely trigger alerts. They surface as a rising DBU line.



— Tuned occasionally — Tuned continuously

Illustrative

03 THE SOLUTION

Lakemine closes the loop, **every day.**

Lakemine turns raw signal into a ranked, applied fix, and repeats it daily so savings hold.

01

Deploy inside the workspace

Installs into your Databricks account and reads system tables, configuration, and telemetry in place.

02

Analyze on a daily cycle

All 17+ engines evaluate compute, SQL, pipelines, storage, and runtime against current demand.

03

Rank by business impact

Each finding carries an estimated saving, so the highest-value work rises to the top.

04

Route to the owning engineer

Findings land with the team that owns the workload, with the evidence behind them.

Runs inside your environment

Execution stays in your Databricks account, which clears the security review that blocks external tools.

Continuous rather than point in time

A daily cycle catches drift as it appears, so savings hold through the quarter.

Databricks-specific by design

Engines are built around Databricks failure modes: DBU spikes, Photon eligibility, spill, autoscaling, cluster policy.

Prioritized by business impact

Every recommendation carries an estimated saving, so a team with limited hours knows what to apply first.

Zero data movement

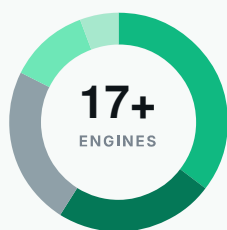
Customer data, notebooks, SQL, and metadata never leave your environment. Private Link supported.

04 THE PLATFORM

17+ built-in optimization engines

Each engine owns one failure mode and runs daily, so configuration, performance, and demand stay covered as things change.

- 01 SQL Warehouse Rightsizer
- 02 All-Purpose Cluster Tuner
- 03 Automated Job Cluster Optimizer
- 04 DLT Pipeline Streamliner
- 05 Serverless Compute Tracker
- 06 Intelligent Autoscaling Assessor
- 07 Runtime Version Auditor
- 08 DBU Anomalous Spike Detector
- 09 AI & ML Workload Profiler
- 10 Photonic Transition Analyzer
- 11 Spot Instance Allocator
- 12 Cluster Policy Guardrail
- 13 Idle Compute Terminator
- 14 Tagging & Metadata Auditor
- 15 Storage Tiering Planner
- 16 Queue & Concurrency Balancer
- 17 Spill-to-Disk Eliminator



ENGINES BY DOMAIN

- 6 Compute & clusters
- 4 SQL & query execution
- 4 Governance & cost control
- 2 Pipelines & AI workloads
- 1 Storage

Covers SQL warehouses, all-purpose and job clusters, DLT pipelines, serverless, Photon, autoscaling, spot instances, and AI/ML workloads.

05 BUSINESS OUTCOMES

What your teams get

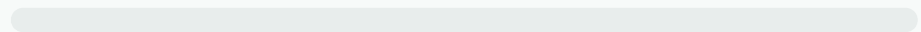
Results compound across finance, platform, and engineering as the cycle keeps the environment tuned.

- ✓ Reduced DBU consumption and less idle resource overhead
- ✓ Higher cluster utilization and cleaner configurations
- ✓ Faster SQL Warehouse and heavy-workload execution
- ✓ Stronger governance and tagging compliance across workspaces
- ✓ Engineering hours returned to delivery work
- ✓ Predictable spend as new teams and AI workloads come online

20–35%

AVERAGE COST REDUCTION

Spend today

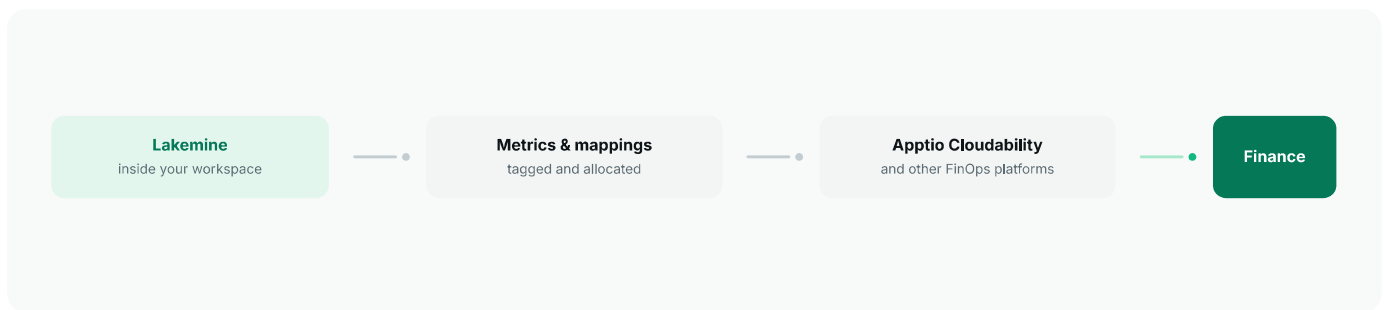


Running Lakemine



Optimization data where **cost decisions happen.**

Lakemine plugs into **IBM Apptio Cloudability** and other platforms, extending the FinOps stack your teams already run.



Native integration

Direct connection to Apptio Cloudability with automated tagging validation and unit-cost allocation.



Executive-ready exports

Summarized metrics, Business Mappings, and chargeback detail, ready for leadership review.



Chargeback by owner

Cost allocated to the team or business unit that drove it, so accountability sits with the owner.



Deployed under your controls

Runs inside your account with Private Link support. The FinOps connection carries no customer data.

07 COMMERCIAL MODELS

Pricing aligned to consumption

Two ways to buy. Choose the one that matches how your finance team plans spend.

MODEL 01

RISK-FREE

Pay as you Save

Pay nothing up front. Billing is based on audited savings, invoiced monthly.

MODEL 02

Pay per DBU

A predictable licence scaled against active DBU consumption, built for flat-rate budgeting.

Find the waste in your Databricks environment.

Lakemine runs a full analysis cycle inside your account and reports what it finds before you commit.

[Book a workspace review](#)

Where Databricks spend **actually** goes

Databricks bills in two currencies. Databricks Units price the platform itself, while the underlying VMs, storage, and networking are billed separately by AWS, Azure, or GCP. A cluster that looks cheap on one line can be expensive on the other, which is why spend analysis that reads only one of them tends to mislead.

Compute is where the money is

Most Databricks bills are dominated by compute: SQL warehouses serving BI, job clusters running scheduled pipelines, all-purpose clusters for interactive work, and increasingly serverless and AI workloads. Storage matters, but it rarely moves the number the way a misconfigured warehouse does.

Six places waste hides

Idle time. Clusters kept warm for convenience, or auto-termination windows set generously and never revisited. The workload finished; the meter did not.

Overprovisioning. Node types and worker counts chosen for a worst-case job that ran once, then left in place as the default for everything else.

Autoscaling misconfiguration. Minimum worker counts set too high, or scaling ranges so wide the cluster thrashes. Autoscaling only saves money when its bounds reflect real demand curves.

Spill to disk. Memory-bound stages that spill run slower and longer, so the same job consumes more DBUs than it needs to. This is invisible to infrastructure monitoring, which sees a healthy host.

Stale runtimes. Older Databricks runtimes miss engine improvements, and workloads eligible for Photon that never moved to it keep paying in execution time.

Unallocated spend. Without disciplined tagging, no one can say which team, product, or model drove the increase, so no one owns reducing it.

Why point-in-time tuning does not hold

Each of these is fixable. The difficulty is that a Databricks environment is not static. New teams onboard, pipelines are added, models are retrained, and query patterns shift. A configuration that was correct in January is merely plausible by April. This is why a tuning engagement produces a real saving and then watches it erode: the work was correct, but it was a snapshot.

How Lakemine closes the loop

Lakemine treats optimization as an operating cycle rather than a project. It deploys inside the customer's Databricks account, reads the environment in place, and runs a full analysis pass every day.

Reading the environment

The engines draw on system tables, cluster and warehouse configuration, job history, and workload telemetry that Databricks already produces. Nothing is copied out, and no agent inspects customer data. The inputs are operational metadata: how compute was configured, what it ran, how long it took, and what it consumed.

Diagnosis, not alerting

An alert says a number moved. A diagnosis says why. Each engine owns a specific failure mode and evaluates the evidence for it: whether a warehouse is oversized for its concurrency, whether a job spilled, whether autoscaling bounds match observed demand, whether a workload is eligible for Photon, whether a runtime is behind.

Ranking by consequence

A long list of findings is its own kind of failure. Every recommendation carries an estimated saving and a performance effect, so a platform team with a few hours a week can work top-down and know the first fix is the one that pays most.

Ownership

Findings are routed to the team that owns the workload, with the supporting evidence attached. Cost is allocated to the group that generated it, which is what turns a shared bill into an accountable one.

Security posture

Execution stays inside the customer's Databricks account. Customer data, notebooks, SQL, and metadata do not leave the environment, and Private Link is supported where network policy requires it. For most security teams this is the difference between a review that takes months and one that does not.

A note on the numbers. The 20–35% range quoted in this document reflects infrastructure cost reduction observed across enterprise deployments. Savings depend on the starting configuration of the environment. A workspace that has already been tuned recently will have less headroom than one that has grown without review. A workspace assessment establishes the actual figure before any commitment.