

# Efficiency Score

Efficiency Score - Cost Analysis body { font-family: Arial, sans-serif; font-size: 11pt; line-height: 1.5; color: #333; margin: 2cm; } .header { border-bottom: 1px solid #ddd; padding-bottom: 15px; margin-bottom: 20px; } .footer { margin-top: 30px; border-top: 1px solid #ddd; padding-top: 10px; font-size: 9pt; color: #666; text-align: center; } h1 { color: #0066cc; font-size: 20pt; margin: 0 0 10px 0; } h2 { color: #0066cc; font-size: 16pt; margin: 20px 0 10px 0; } h3 { font-size: 13pt; margin: 15px 0 10px 0; } .meta { color: #666; font-size: 10pt; } table { width: 100%; border-collapse: collapse; margin: 15px 0; font-size: 10pt; } th { background-color: #f1f3f8; border: 1px solid #ddd; padding: 8px; text-align: left; } td { border: 1px solid #ddd; padding: 8px; } Efficiency Score Cost Analysis Report - Generated on April 23, 2025 Based on the dashboard data and provided analysis, here are some cost optimization recommendations for Databricks usage: 1.

All-Purpose Compute Optimization: Recommendation: Right-size clusters and leverage autoscaling for the Payments business unit.

Expected Impact: Significant cost reduction due to reduced wasted compute resources.

Potentially improved performance by eliminating resource contention if over-provisioning is causing data spills.

Implementation Difficulty: Medium Specific Steps: Analyze cluster utilization metrics (CPU, memory, disk I/O) for Payments' all-purpose clusters.

Identify peak and off-peak usage patterns.

Configure autoscaling policies to dynamically adjust cluster size based on demand.

Start with conservative scaling parameters and adjust as needed.

Choose smaller instance types for driver and worker nodes if analysis shows consistent over-provisioning, even during peak periods.

Monitor cluster performance after implementing changes and fine-tune as needed.

Further Investigation: Investigate data spills for Payments to confirm if over-provisioning is leading to performance degradation despite a high efficiency score.

2.

Job Compute Optimization: Recommendation: Address missing data for Stablecoin and Marketing, then right-size Finance's job clusters.

Expected Impact: Moderate cost reduction for Finance, improved data reliability.

Implementation Difficulty: Medium Specific Steps - Missing Data: Verify that Stablecoin jobs are actually running and that data collection is configured correctly.

Check job logs and Databricks monitoring tools.

Investigate why Marketing has intermittent missing data.

Check job schedules and data pipelines.

Specific Steps - Right-sizing Finance: Analyze cluster configurations for Finance's jobs.

Determine appropriate instance types and cluster sizes based on job requirements and historical resource usage.

Implement changes and monitor job performance.

Note: The high memory scores for Finance suggest over-provisioning.

Address this after resolving data collection issues to ensure accurate assessment.

3.

SQL Warehouse Optimization: Recommendation: Optimize SQL queries for Ripplex and investigate the efficiency spike for Technical Services.

Investigate missing data for Finance.

Expected Impact: Moderate to significant cost reduction depending on the optimization potential of identified queries.

Implementation Difficulty: Medium to High (depending on query complexity) Specific Steps: Analyze Ripplex's SQL queries to identify performance bottlenecks.

Use query profiling tools available in Databricks.

Optimize queries by rewriting inefficient code, leveraging caching where appropriate, and optimizing data access patterns.

Investigate the cause of the Technical Services spike on 2024-02-25.

Identify the specific query or workload that caused the spike and optimize it.

Investigate the missing data for Finance in the SQL Warehouse metrics and resolve the data collection issue.

4.

S3 Optimization: Recommendation: Investigate the high efficiency scores for Finance and Compliance on their respective dates.

Improve data localization and partitioning.

Expected Impact: Small to moderate cost reduction and potential performance improvements.

Implementation Difficulty: Medium Specific Steps: Analyze S3 access patterns for Finance and Compliance on the dates with high efficiency scores.

Identify the specific data being accessed and the queries or jobs involved.

Optimize data access by improving data localization (placing data closer to compute resources) and partitioning (organizing data for more efficient retrieval).

Investigate and resolve the cause of any missing data in S3 metrics.

General Recommendations: Cost Monitoring and Alerting: Set up automated alerts for cost thresholds and anomalies to proactively identify potential issues.

Resource Tagging: Tag resources by business unit, project, or environment to gain granular visibility into costs and allocate them appropriately.

Regularly Review Cluster Configurations: Periodically review cluster configurations and right-size them as needed to ensure optimal resource utilization.

Utilize Spot Instances (if applicable): Consider using spot instances for non-critical workloads to further reduce costs.

By implementing these recommendations, you can significantly optimize your Databricks costs while maintaining or even improving performance.

Remember that continuous monitoring and analysis are crucial for sustained cost optimization.

Generated by Grafana Cost Analyzer · Confidential.