# AWS Data Lake Cost Optimization Plan

This document outlines areas identified across AWS Data Lake accounts where cost savings can be achieved. The opportunities are grouped into two categories: infrastructure changes and code-level improvements. These changes are aimed at improving efficiency, reducing unnecessary spend, and building better visibility into how resources are used.

## Infrastructure Changes

### 1. Redshift

* Scheduling Clusters:

Redshift clusters are currently running 24/7, even during nights and weekends when there’s no activity. I plan to implement a scheduling mechanism to pause the clusters during off-hours to save on compute costs.

* Monitoring for Right-Sizing:

I want to set up a monitoring system using CloudWatch metrics and cost data to evaluate Redshift usage. Based on trends, we can suggest downgrading underused clusters or upgrading those that are consistently maxed out.

### 2. Glue

* Enforcing Job Configuration Controls:

Some Glue jobs are using high-cost worker types (like G.2X) or excessive workers without needing them. I plan to introduce guardrails — like IAM restrictions or CI/CD checks — to flag or prevent such configurations unless there's a clear need.

### 3. Cost Usage Reporting for Projects

* Using Cost Usage Reports (CUR):

Cost usage reports are already set up in the accounts. I plan to use them to generate monthly summaries for leadership, showing detailed spend by project. This will help identify which resources are driving costs and where optimizations can be made.

## Code Changes

### 1. Moving PGP Decryption to Lambda

* Current Setup:

Most ETL pipelines begin with a Glue job that only performs PGP decryption — a lightweight task that doesn't require distributed compute.

* Proposed Change:

Replace the decryption Glue step with a Lambda function to handle PGP decryption. This will lower costs and speed up pipeline execution.

### 2. Redshift Load Jobs with Lambda

* Current Setup:

Some pipelines use Glue jobs to load data into Redshift using COPY commands. These jobs can run for hours due to delays caused by lack of available IPs for worker nodes, not because of actual data processing.

* Proposed Change:

Move the Redshift load step to Lambda, which only needs to trigger the COPY command inside Redshift. This avoids provisioning overhead and subnet/IP issues, leading to faster and cheaper loads.

### 3. Exploring Alternatives to Glue for Growing ETL Workloads

* Current Setup:

Glue usage is increasing across projects as datasets grow, driving up costs — especially with multiple pipelines per project.

* Proposed Change:

Explore using EMR clusters (already approved) for larger batch jobs where possible. EMR provides better control over compute resources and pricing flexibility (e.g., spot instances). Once container workloads are approved, we can also explore running containerized ETL jobs using ECS or EKS for even more cost-effective execution.