AWS Anomaly Automation - Architecture Overview

The AWS Anomaly Automation architecture provides a scalable and centralized solution for detecting and tracking cost anomalies across multiple AWS accounts—spanning both production and non-production environments. By leveraging core AWS services such as AWS Lambda, Amazon S3, and a NoSQL database, the system enables secure cross-account integration, real-time anomaly processing, and streamlined data management.

# Workflow Components

## 1. Account Sources

The system integrates with two categories of AWS accounts:  
- np-antiphoney – Non-production AWS accounts  
- antiphoney – Production AWS accounts  
  
These serve as the source for cost and usage data across the environment.

## 2. AWS Lambda

A central AWS Lambda function orchestrates the automation process:  
- Reads account IDs and metadata from Amazon S3  
- Assumes cross-account roles to securely access each AWS account  
- Retrieves temporary security credentials and invokes the AWS Cost Anomaly Detection API

## 3. Amazon S3

Acts as the metadata store, containing:  
- AWS Account IDs  
- Environment-specific tags  
- Role ARNs or access tokens for Lambda authentication

## 4. Amazon DynamoDB / NoSQL Database

Used to persist structured data including:  
- Anomaly details in JSON format  
- Cost breakdowns and associated root cause information

## 5. External API Layer

An optional external API (via API Gateway or AppSync) allows:  
- Access to anomaly records through dashboards or reports  
- Integration with alerting tools or ticketing systems for automated workflows

## 6. AWS Cost Explorer / Anomaly Detection Service

The Lambda function interacts with AWS-native services to fetch:  
- Identified anomalies  
- Impacted services and accounts  
- Affected time windows and cost values

