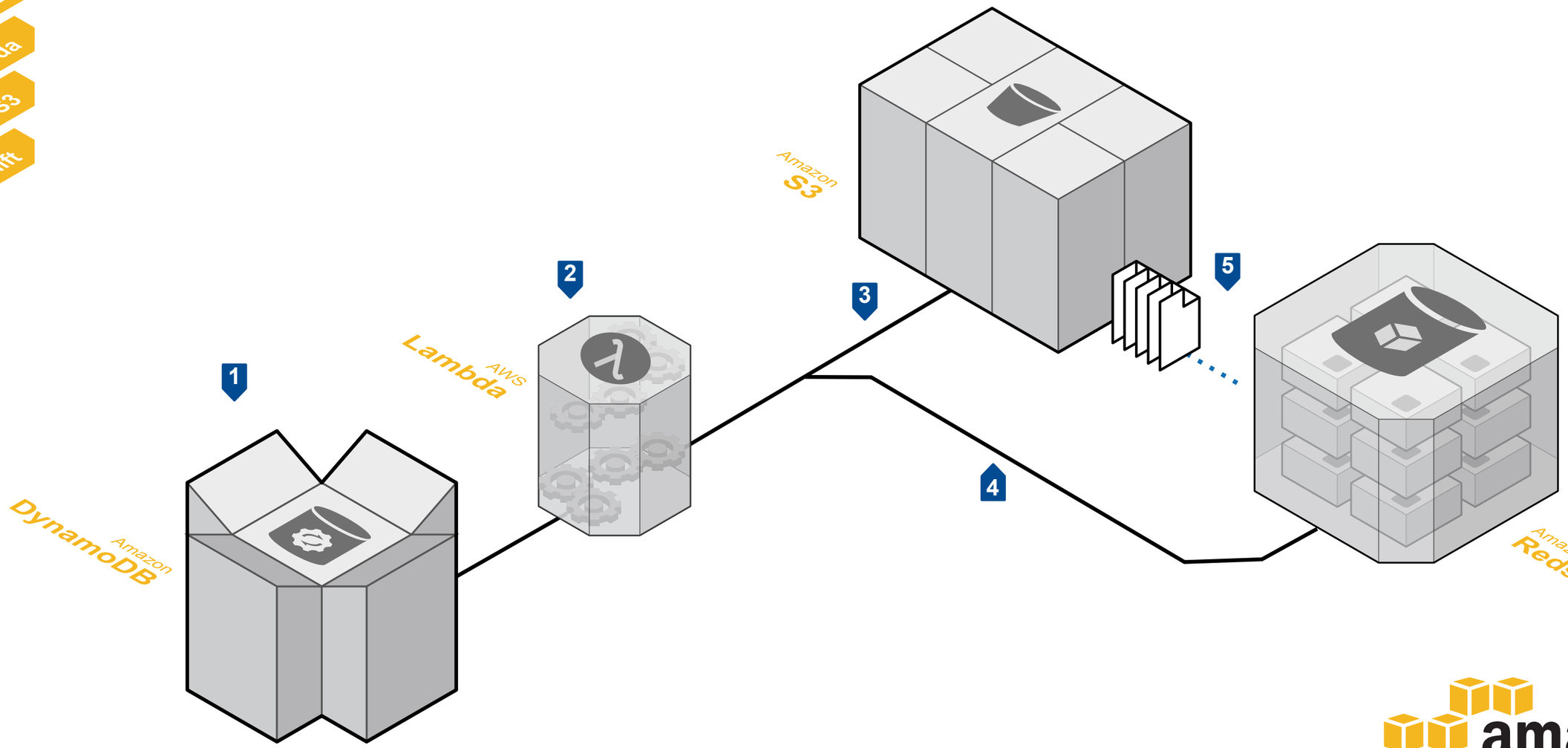
AWS Reference Architectures

- Amazon DynamoDB
- AWS Lambda
- Amazon S3
- Amazon Redshift

AWS LAMBDA: EXTRACT, TRANSFORM, LOAD

Amazon DynamoDB has been designed to provide consistently fast access to data records, with unrivaled performance for accessing individual record sets or collections. Conversely, data sets that have been archived to Amazon Redshift are optimized for complex queries that span multiple sources and huge timespans. This pairing of DynamoDB Streams with AWS Lambda can facilitate a maintenance-free, serverless data warehousing solution that puts the power of both DynamoDB and Amazon Redshift in the hands of your applications and data scientists.



amazon web services

System Overview

1 Data is ingested into a real-time, easily scalable **Amazon DynamoDB** table that is configured to publish updates to a DynamoDB Stream.

2 An AWS Lambda function subscribes to this stream to receive batches of updates from DynamoDB. New batches of updates are transformed into a compressed CSV format, with an opportunity first to perform some data enrichment (e.g., Geo-IP lookups).

3 Lambda uploads the compressed CSV content (other delimiters are also supported) as parts of **Amazon Simple Storage Service (Amazon S3)** multipart uploads, ready to be ingested by **Amazon Redshift**. The files can be left here as a long term source of truth, or Amazon S3 lifecycle policies could archive to Amazon Glacier or delete after some configurable period.

4 The Lambda function periodically calls Amazon Redshift's COPY command to perform an import of newly accumulated CSV files from Amazon S3. When the Lambda function calls Amazon Redshift's COPY command, Amazon Redshift distributes the CSV file retrieval and ingest tasks across the cluster.

5 Amazon Redshift retrieves the most recent content from Amazon S3 and then distributes the ingest files across each cluster node for parallelized imports.