1. **AWS Cost and Usage Reports at scale**

**Problem:**

A company has multiple accounts under the same organization and it uses a consolidated billing to follow up on the overall costs. AWS provides a report in CSV format of all detailed costs. This report can be send to bucket of your S3, as defined in the Cost and Usage reports documentation: <https://docs.aws.amazon.com/cur/latest/userguide/cur-create.html>.

The company wants to provide useful and organized data on the costs of your infrastructure, thus facilitating resource optimization. The company's AWS usage is quickly growing and the stakeholders are expecting near real-time insights to better support their planing and budget allocation. The body of stakeholders is formed by both technical and non-technical professionals, composed by engineers, product owners and high level management.

**Solution:**

[AWS Cost and Usage Reports (AWS CUR)](https://docs.aws.amazon.com/cur/latest/userguide/what-is-cur.html) are helpful for customers who want to optimize their AWS spending, allocate costs internally, and drive the right governance model for their organizations.

Our solution extends this functionality by providing automation that enables delivery of CUR reports directly to the stakeholders to support their planning and adjustments.

The solution works by configuring the CUR report to S3 bucket and then used AWS Glue for ETL operation, final result will be stored in Amazon Redshift for the reporting service.

The following diagram depicts the CUR report automation process flow.

Graphical user interface, application, Teams

Description automatically generated

Configure CUR Report to S3 bucket:

* Create S3 bucket postnl.(Bucket name: postnlbusket)
* Configure to CUR report on hourly basis to store the report into S3 bucket.

Graphical user interface, text, application, email

Description automatically generated

Create Target S3 bucket:

Cdk code:

aws\_lambda/s3\_stack/s3bucket.py

AWS Glue ETL Operation:

* Create glue database connection and table using the following stack.

Cdk code path:

aws\_lambda/glue/glue\_etl\_db\_table.py

* Create glue job for running the transformation script in aws glue

ETL transformation script:

aws\_lambda/glue/etl\_script/glue\_script\_cur.py

Cdk code path:

aws\_lambda/glue/glue\_etl\_job.py

* Create crawler for updating the table Schema

Cdk code path:

aws\_lambda/glue/glue\_etl\_crawler.py

Lambda function to Trigger Glue Job:

Cdk Code path:

aws\_lambda/aws\_lambda /aws\_lambda\_stack.py

Reporting Solution:

Once ETL Glue job completed, the final transformation output will be loaded in s3 target location.

We can analyse the report using Amazon Athena.

Orelse we will load the s3 transformed output data in Amazon Redshift and the we can configure to Power BI report.