**Airflow DAG3 Document**

In DAG3, we implemented the development of an automated email notification system utilizing Apache Airflow, Google Cloud Pub/Sub, Google Cloud Functions, and SendGrid along with updating of the latest predictions view on Big Query for the dashboard. Below is a detailed description of each task:

**1. Query\_bigquery Task:** This task queries Google BigQuery to retrieve the latest predictions about customers. It specifically targets the most recently created table in a dataset, looking for customers predicted to churn, who have been using their equipment for more than 340 days.

**How it Works**:

* Initializes a BigQuery client.
* Executes a query to identify the latest table where the prediction results are stored.
* Runs another query on this table to fetch details such as CustomerID, EmailID, and CurrentEquipmentDays for customers meeting the specified criteria.
* The results are stored using XComs for use by subsequent tasks.

**2. Update\_latest\_view Task:** Updates a BigQuery view to reflect the latest prediction data across all tables prefixed with 'predictions\_' in the dataset. This ensures that downstream processes always have access to the most current data in an aggregated format.

**How it Works**:

* Initializes a BigQuery client.
* Dynamically constructs and executes a SQL script to aggregate all prediction tables into a single view, handling schema evolution and ensuring the view always represents the latest data snapshot.
* Uses BigQuery's scripting capabilities (DECLARE, SET, EXECUTE IMMEDIATE) to handle dynamic SQL execution.

**3. Send\_pubsub\_messages Task:** Publishes messages to a specified Google Cloud Pub/Sub topic. Each message contains details about a customer who meets the churn prediction and equipment usage criteria, intended for downstream processing or notification systems.

**How it Works**:

* Pulls data from XComs that was pushed by the query\_bigquery task.
* Initializes a Pub/Sub publisher client.
* Iterates over the data, creating and sending a message for each record. Each message includes the customer's ID, email, and equipment usage days.
* Handles errors during message publication and logs the outcome of each message dispatch.

In addition to the Dag file code, there were some additional setup and configuration to be made:

**1. IAM Role Configuration**

Ensured proper permissions for accessing BigQuery and Pub/Sub:

* Assigned BigQuery Data Viewer and BigQuery Metadata Viewer roles to the Airflow service account.
* Assigned Pub/Sub Publisher role to enable message publishing.

**2. Pub/Sub Integration**

* Created a Pub/Sub topic named customer\_notifications manually through the Google Cloud Console to serve as a messaging channel.
* The Airflow DAG publishes messages to this topic, containing the necessary customer details for each identified recipient.

A screenshot of a computer

Description automatically generated

**3. Email Dispatch with Google Cloud Functions**

Tried to automate the sending of emails without scaling or managing server infrastructure.

* Deployed a Python-based Cloud Function through the console, subscribing to the customer\_notifications Pub/Sub topic.
* Upon triggering by message arrival, the function uses SendGrid to send an email to the customer. The content of the email includes a reminder to upgrade their equipment.

A screenshot of a computer

Description automatically generated

**4. Email Service Integration with SendGrid**

SendGrid reliably send emails with high deliverability.

* Created a SendGrid account, and an API key is generated for secure API calls.
* The API key is stored securely in the environment variables of the Google Cloud Function to ensure security and compliance.
* Configured the SendGrid service to send emails from a designated no-reply sender address, using the API for programmatically sending mail to the fetched customer emails.

However, the send grid API key wasn’t properly connecting and was unable to send emails through send grid. It is something we’d like to keep in future scope.