Data & AI Boot-Kon Event

Title: ELT & Lakehouse : BigQuery, Dataform, LLM for sentiment analysis

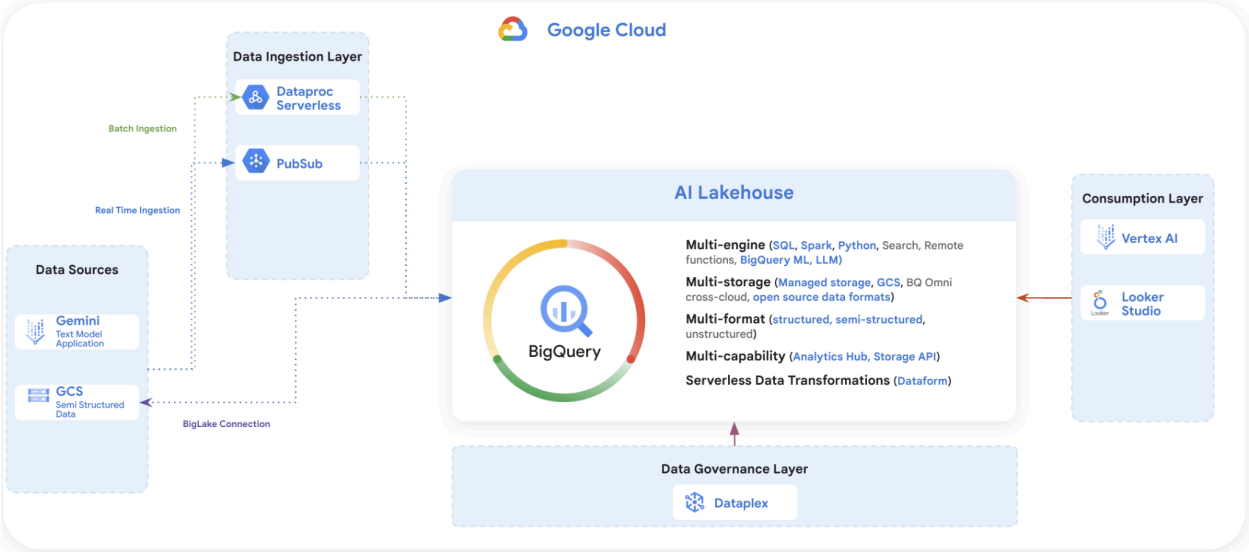
Goal of the lab

- Use Dataform for SQL Transformations in BigQuery.
- Perform sentiment analysis of customer feedback.

Author: Wissem Khlifi **Date**: 2024-04-01 **Estimated Completion Time**: 45 Minutes

CAUTION:
This lab is for educational purposes only and should be used with caution in production environments. Google Cloud Platform (GCP) products are changing frequently, and screenshots and instructions might become inaccurate over time. Always refer to the latest GCP documentation for the most up-to-date information.

Architecture Diagram:



ELT: Dataform & LLM for sentiment analysis from BigQuery

READING Section : Read the following Explanation of what dataform is and what is its purpose

Dataform

Dataform is a fully managed service that helps data teams build, version control, and orchestrate SQL workflows in BigQuery. It provides an end-to-end experience for data transformation, including:

- Table definition: Dataform provides a central repository for managing table definitions, column descriptions, and data quality assertions. This makes it easy to keep track of your data schema and ensure that your data is consistent and reliable.

- Dependency management: Dataform automatically manages the dependencies between your tables, ensuring that they are always processed in the correct order. This simplifies the development and maintenance of complex data pipelines.
- Orchestration: Dataform orchestrates the execution of your SQL workflows, taking care of all the operational overhead. This frees you up to focus on developing and refining your data pipelines.

Dataform is built on top of Dataform Core, an open source SQL-based language for managing data transformations. Dataform Core provides a variety of features that make it easy to develop and maintain data pipelines, including:

- Incremental updates: Dataform Core can incrementally update your tables, only processing the data that has changed since the last update. This can significantly improve the performance and scalability of your data pipelines.
- Slowly changing dimensions: Dataform Core provides built-in support for slowly changing dimensions, which are a common type of data in data warehouses. This simplifies the development and maintenance of data pipelines that involve slowly changing dimensions.
- Reusable code: Dataform Core allows you to write reusable code in JavaScript, which can be used to implement complex data transformations and workflows.

Dataform is integrated with a variety of other Google Cloud services, including GitHub, GitLab, Cloud Composer, and Workflows. This makes it easy to integrate Dataform with your existing development and orchestration workflows.

Benefits of using Dataform in Google Cloud

There are many benefits to using Dataform in Google Cloud, including:

- Increased productivity: Dataform can help you to increase the productivity of your data team by automating the development, testing, and execution of data pipelines.
- Improved data quality: Dataform can help you to improve the quality of your data by providing a central repository for managing table definitions, column descriptions, and data quality assertions.
- Reduced costs: Dataform can help you to reduce the costs associated with data processing by optimizing the execution of your SQL workflows.
- Increased scalability: Dataform can help you to scale your data pipelines to meet the needs of your growing business.

Use cases for Dataform

Dataform can be used for a variety of use cases, including:

- Data warehousing: Dataform can be used to build and maintain data warehouses that are scalable and reliable.
- Data engineering: Dataform can be used to develop and maintain data pipelines that transform and load data into data warehouses.
- Data analytics: Dataform can be used to develop and maintain data pipelines that prepare data for analysis.
- Machine learning: Dataform can be used to develop and maintain data pipelines that prepare data for machine learning models.

LAB Section : Dataform Prerequisites

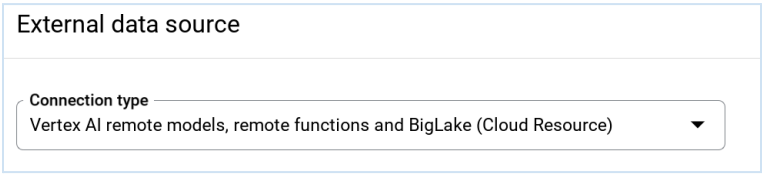
1. **Enable Services API (you can skip this step if you completed LAB 1)**

Ensure all necessary APIs (BigQuery API, Vertex AI API, BigQuery Connection API, Dataform API, Secret Manager API) are enabled

2. **Create a connection to an external data source in BigQuery (you can skip this step if you completed LAB 1)**

- Create an External Connection (Enable BQ Connection API if not already done) and note down the Service Account id from the connection configuration details:
- Click the +ADD button on the BigQuery Explorer pane (in the left of the BigQuery console) and click “Connection to external data sources” in the popular sources listed

- Select Connection type as “Vertex AI remote models , remote functions and Biglake” and provide “fraud-transactions-conn” as Connection ID, select Multi Region location type.



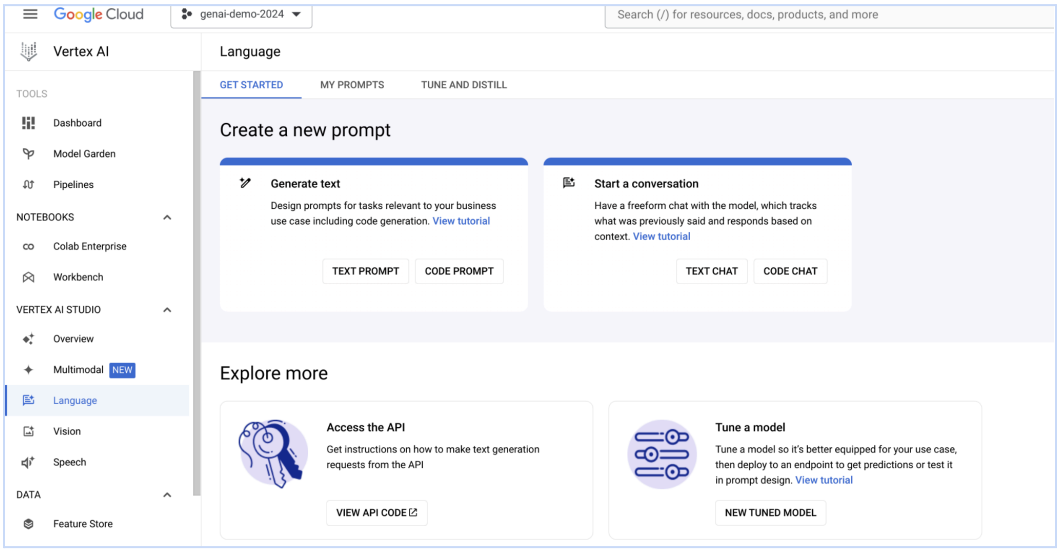
- Once the connection is created, take a note of the Service Account generated from the connection configuration details

3. Grant Permissions (you can skip this step if you completed LAB 1)

In this step we will grant permissions to the Service Account to access the Vertex AI service:
Open IAM and add the Service Account you copied after creating the external connection as the Principal and select “Vertex AI User” Role.

4. Using Large Language Models from Vertex AI

Google Cloud’s language models are available within the Vertex AI Studio inside the Vertex AI service.



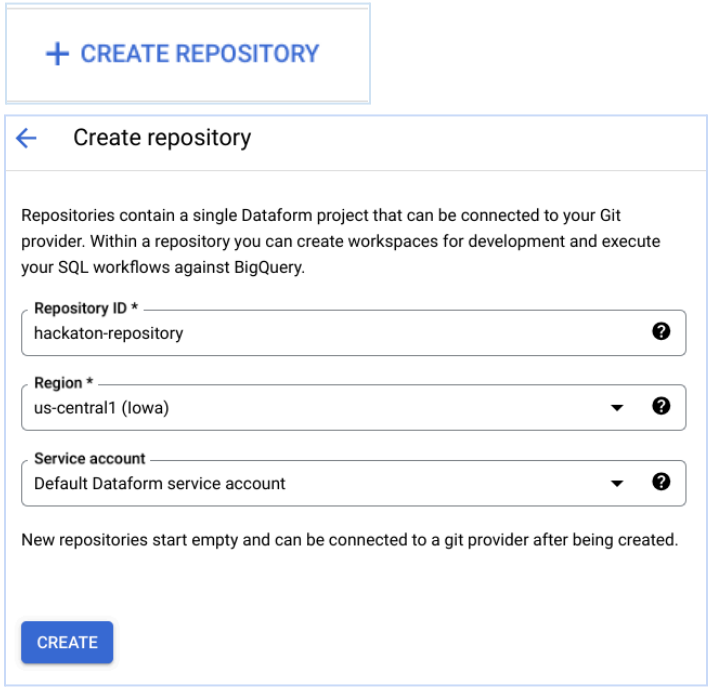
5. Prompt design

Prompt design is the process of creating prompts that elicit the desired response from language models. Writing well structured prompts is an essential part of ensuring accurate, high quality responses from a language model. If you need to understand this concept a bit more this is a page that introduces some basic concepts, strategies, and best practices to get you started in designing prompts (https://cloud.google.com/vertex-ai/docs/generative-ai/learn/introduction-prompt-design).
The reference page above also goes into the more advanced settings you can see on the right hand side of the prompt box such as temperature, top K, top P etc.

LAB Section : Creating a Dataform Pipeline

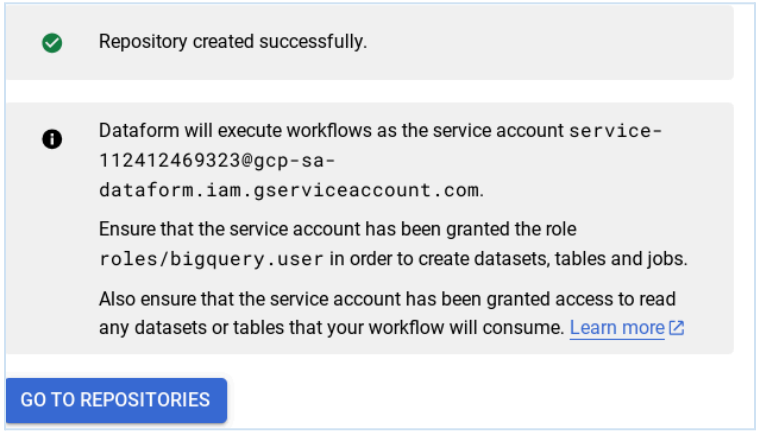
First step in implementing a pipeline in Dataform is to set up a repository and a development environment. Detailed quickstart and instructions can be found here.

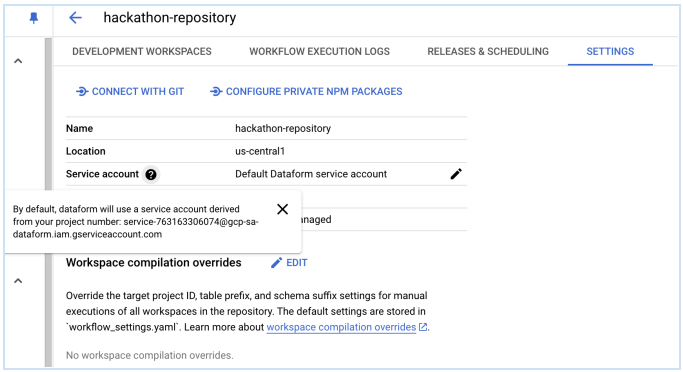
1. Create a Repository in Dataform



2. Dataform Service Account

Take note and save somewhere the newly created service account for Dataform.
Example: service-112412469323@gcp-sa-dataform.iam.gserviceaccount.com





3. Create and initialize a Dataform development workspace

a. In the Google Cloud console, go to the **Dataform** page.
Go to Dataform
b. Click on the new repository you have just created.
c. Click add **Create development workspace**.
d. In the **Create development workspace** window, do the following:
  a. In the **Workspace ID** field, enter "hackathon-<YOURLASTNAME>-workspace" (replace <YOURLASTNAME> with your name)
  b. Click **Create**.
e. The development workspace page appears.
f. Click on the newly created development workspace
g. Click **Initialize workspace**.

h. You will copy the dataform files from the following repository, in the next steps.

https://github.com/dace-de/bootkon-h2-2024/tree/main/dataform

i. *Edit* ***workflow_settings.yaml*** *file :*

- *Replace defaultDataset value with* ***ml_datasets*** **,** make sure defaultProject value should be ***your project id***

Note: Nevermind if you have a different dataform core version, just continue

workflow_settings.yaml

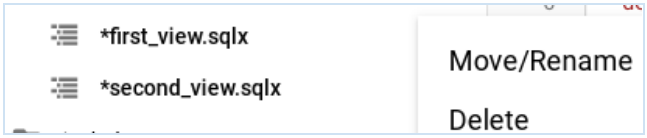
```
defaultProject: genai-demo-2024
defaultLocation: US
defaultDataset: ml_datasets
defaultAssertionDataset: dataform_assertions
dataformCoreVersion: 3.0.0-beta.4
```

- Click on Install Packages ***Only Once***.

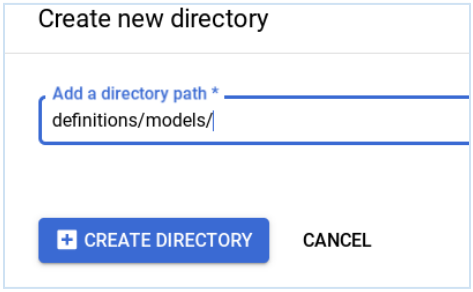
Package installation succeeded

j. *Remove from your local dataform repository the following SQLX files; Delete the following files;*

- *first_view.sqlx*
- *second_view.sqlx*



k. *Click on definitions and create a new directory called "models":*

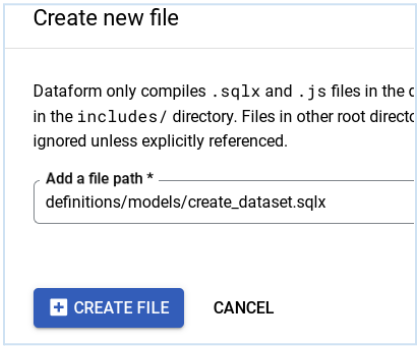


l. *Click on models directory and create 2 new files ; (make sure all file names are in lowercase)*

- create_dataset.sqlx
- llm_model_connection.sqlx

Those files should be created under ***definitions/models*** directory

Example:



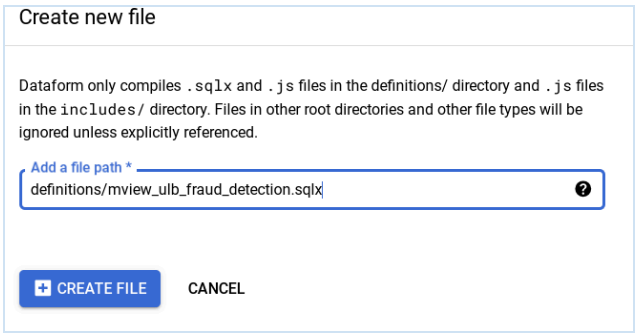
m. *Copy the contents from https://github.com/dace-de/bootkon-h2-2024/tree/main/dataform/definitions/models to each of those files.*

n. *Click on definitions and create 3 new files: (make sure all file names are in lowercase)*

- mview_ulb_fraud_detection.sqlx
- sentiment_inference.sqlx
- ulb_fraud_detection.sqlx

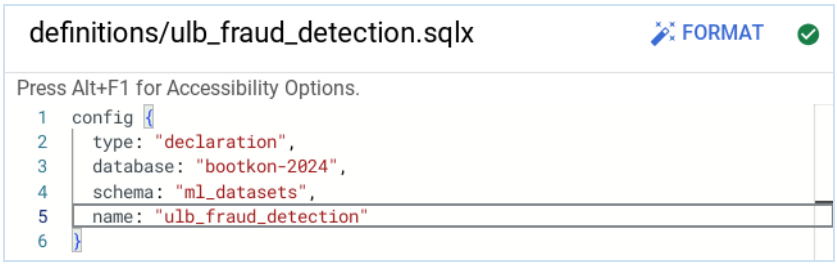
Those files should be created under ***definitions*** directory

Example:



o. *Copy the contents from https://github.com/dace-de/bootkon-h2-2024/tree/main/dataform/definitions to each of those files.*

p. *Make sure to replace database by your project ID value in ulb_fraud_detection.sqlx file*

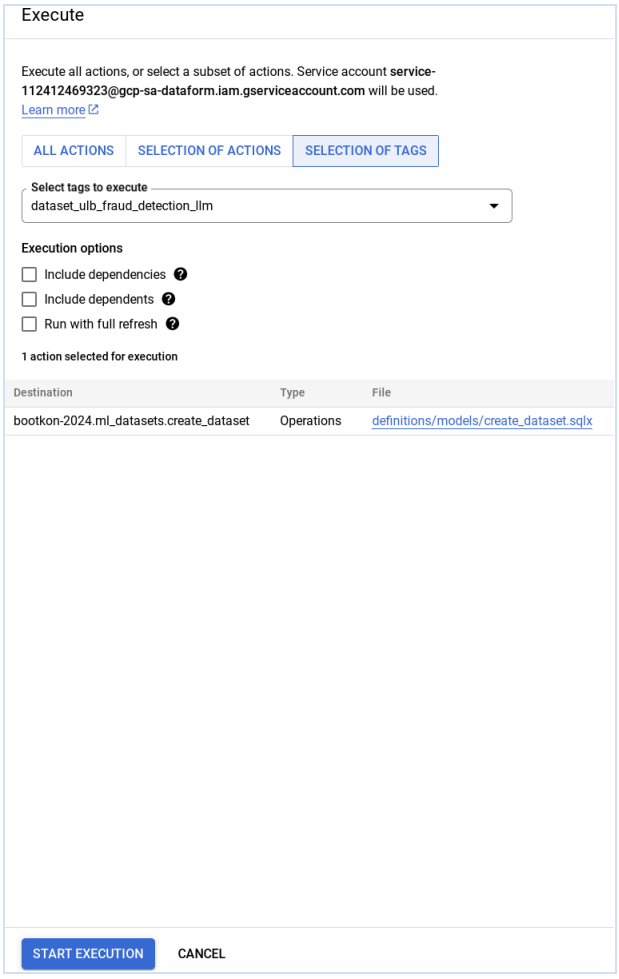


q. *In llm_model_connection.sqlx, replace the `us.llm-connection` connection with the connection name you have created in LAB 2 during the BigLake section. If you have followed the steps in LAB 2, the connected name should be "**us.fraud-transactions-conn**"*

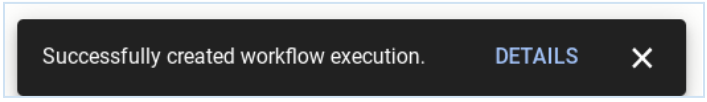
Notice the usage of $ref in line 10, of ***definitions/mview_ulb_fraud_detection.sqlx*** "sqlx" file. The advantages of using $ref in Dataform are

- Automatic Reference Management: Ensures correct fully-qualified names for tables and views, avoiding hardcoding and simplifying environment configuration.
- Dependency Tracking: Builds a dependency graph, ensuring correct creation order and automatic updates when referenced tables change.
- Enhanced Maintainability: Supports modular and reusable SQL scripts, making the codebase easier to maintain and less error-prone.

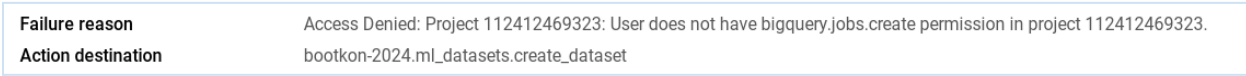
5. *Run the dataset creation by TAG. TAG allows you to just execute parts of the workflows and not the entire workflow. Click on Start Execution > Tags > "dataset_ulb_fraud_detection_llm" > Start Execution*



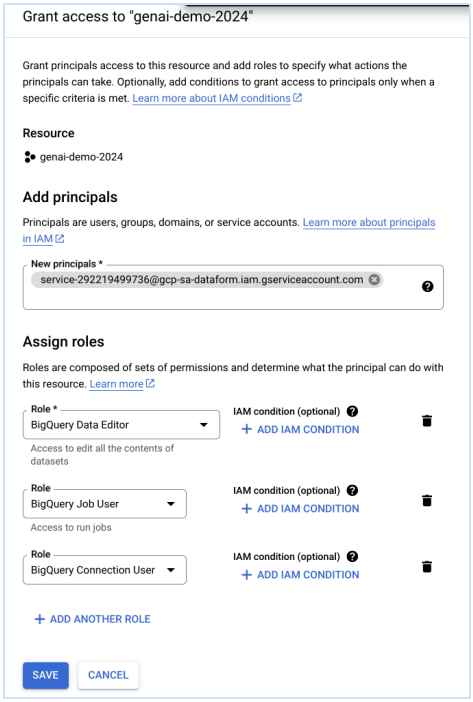
6. *Click on Details;*



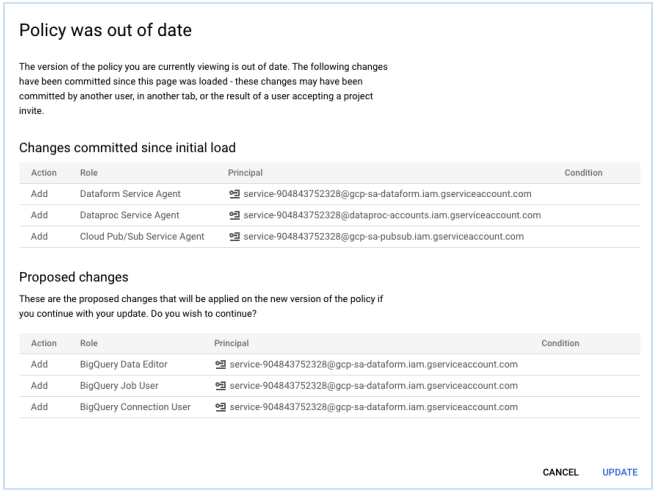
7. *Notice the Access Denied error on BigQuery for the dataform service account XXX@gcp-sa-dataform.iam.gserviceaccount.com;*



8. Go to IAM & Admin > Grant access and grant ***BigQuery Data Editor , BigQuery Job User and BigQuery Connection User*** to the data from the service account. Click on Save.



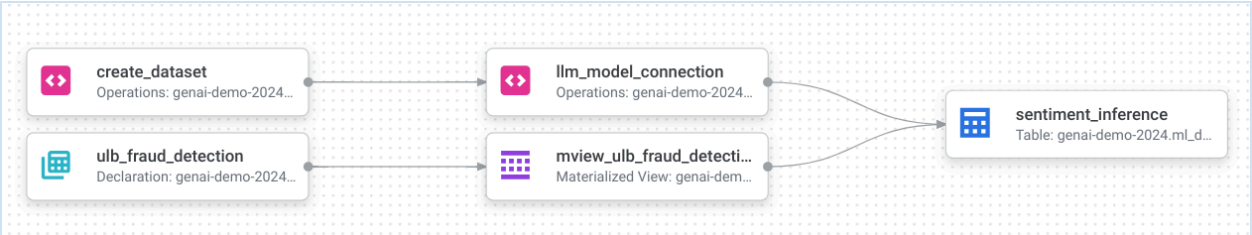
Note: If you encounter the following policy update screen, just click on update.



9. Go back to dataform from the BigQuery console, and retry step ***5***. Notice the execution status. It should be a success.

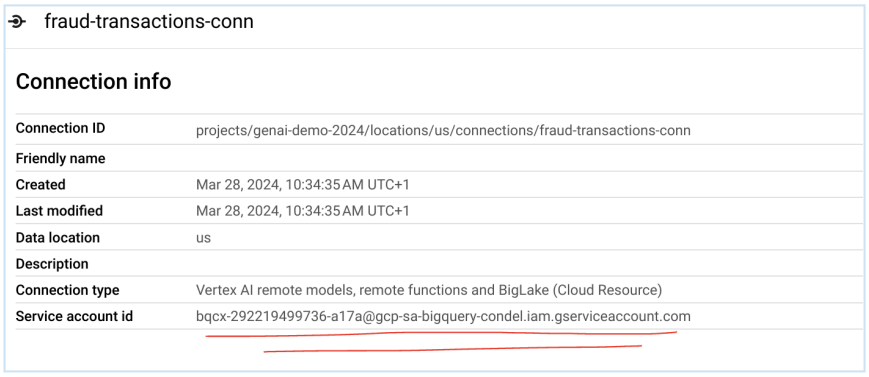
10. Click on Compiled graph and explore it;

Go to ***Dataform > Compiled Graph***



LAB Section : Execute the workspace workflow

1. For the sentiment inference step to succeed . You need to grant the external connection service account the Vertex AI user privilege. More details can be found in this link.

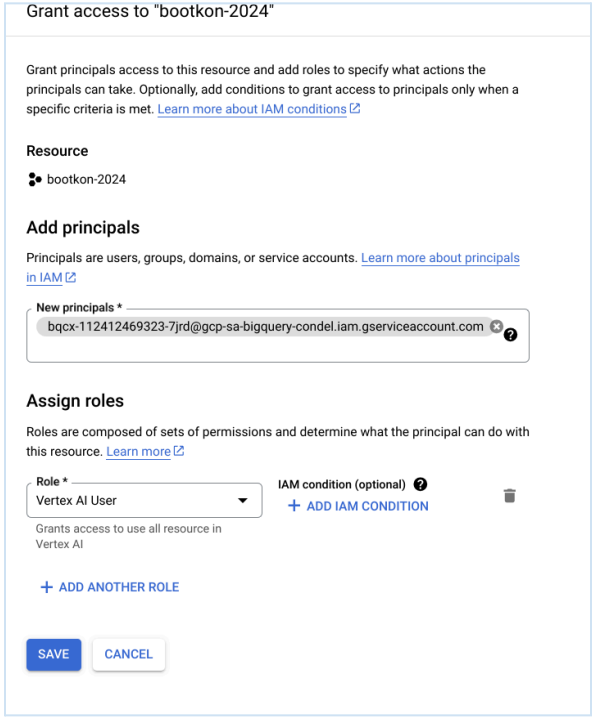


fraud-transactions-conn

Connection info

Connection ID	projects/genai-demo-2024/locations/us/connections/fraud-transactions-conn
Friendly name	
Created	Mar 28, 2024, 10:34:35 AM UTC+1
Last modified	Mar 28, 2024, 10:34:35 AM UTC+1
Data location	us
Description	
Connection type	Vertex AI remote models, remote functions and BigLake (Cloud Resource)
Service account id	bqcx-292219499736-a17a@gcp-sa-bigquery-condel.iam.gserviceaccount.com

2. Take note of the service account and grant it the ***Vertex AI User*** role.



Grant access to "bootkon-2024"

Grant principals access to this resource and add roles to specify what actions the principals can take. Optionally, add conditions to grant access to principals only when a specific criteria is met. Learn more about IAM conditions

Resource

bootkon-2024

Add principals

Principals are users, groups, domains, or service accounts. Learn more about principals in IAM

New principals *

bqcx-112412469323-7jrd@gcp-sa-bigquery-condel.iam.gserviceaccount.com

Assign roles

Roles are composed of sets of permissions and determine what the principal can do with this resource. Learn more

Role * Vertex AI User

Grants access to use all resource in Vertex AI

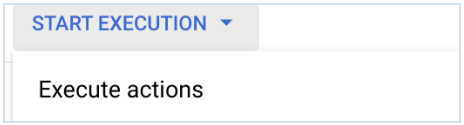
IAM condition (optional)

+ ADD IAM CONDITION

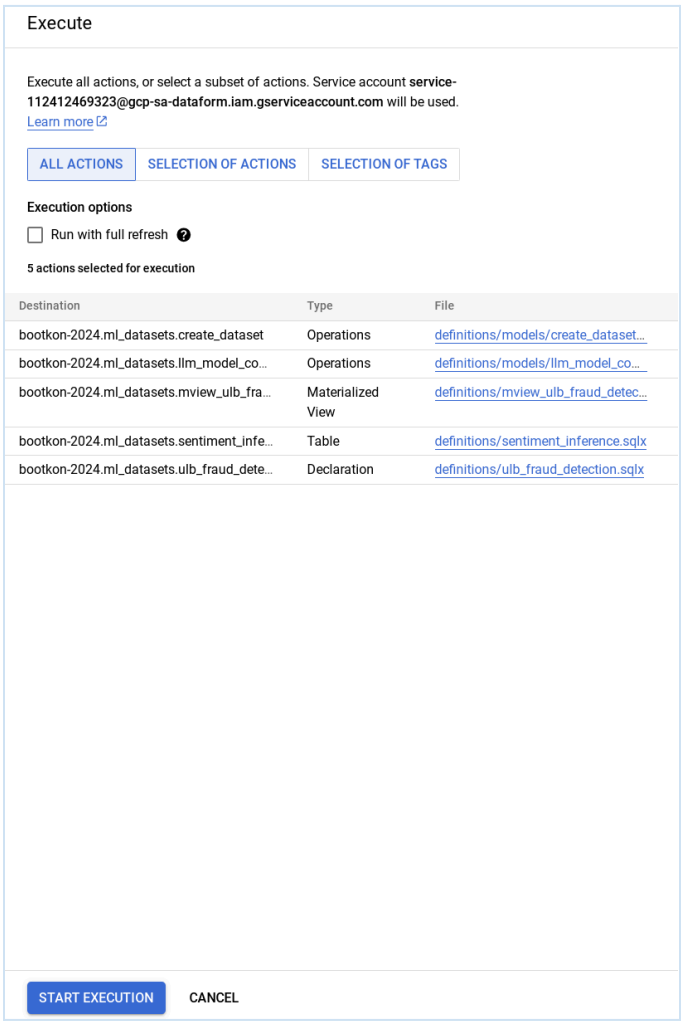
+ ADD ANOTHER ROLE

SAVE CANCEL

3. *Click **START EXECUTION** from the top menu, then* **"Execute actions"**.



4. Click on ***ALL ACTIONS*** Tab then Click on ***START EXECUTION***



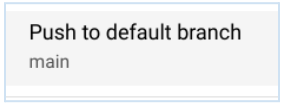
5. Check the execution status. It should be a success.
6. Verify the new table **sentiment_inference** in the ml_datasets dataset in BigQuery.
7. Query the BigQuery table content (At this point you should be familiar with running BigQuery SQL)

BigQuery SQL : Check few rows of **sentiment_inference table**
Replace the project-id by your project id value.

```
SELECT distinct ml_generate_text_llm_result,
prompt,
Feedback
FROM `project-id.ml_datasets.sentiment_inference` LIMIT 10;
```

8. **[Max 2 minutes]** Discuss the table results within your team group.

9. Before moving to the challenge section of the Lab, make sure to ***"Commit X Changes"*** (X should be about 7) and then ***"Push to Default Branch"***



You should now have the message

CHALLENGE Section : Production, Scheduling and Automation

Automate and schedule the compilation and execution of the pipeline. This is done using release configurations and workflow configurations.

Release Configurations:
Release configurations allow you to compile your pipeline code at specific intervals that suit your use case. You can define:

- Branch, Tag, or Commit SHA: Specify which version of your code to use.
- Frequency: Set how often the compilation should occur, such as daily or weekly.
- Compilation Overrides: Use settings for testing and development, such as running the pipeline in an isolated project or dataset/table.

Common practice includes setting up release configurations for both test and production environments. For more information, refer to the release configuration documentation.

Workflow Configurations

To execute a pipeline based on your specifications and code structure, you need to set up a workflow configuration. This acts as a scheduler where you define:

- Release Configuration: Choose the release configuration to use.
- Frequency: Set how often the pipeline should run.
- Actions to Execute: Specify what actions to perform during each run.

The pipeline will run at the defined frequency using the compiled code from the specified release configuration. For more information, refer to the workflow configurations documentation.

[TASK] Challenge : Take up to 10 minutes to Setup a Daily Frequency Execution of the Workflow

Goal: Set up a daily schedule to automate and execute the workflow you created.

1. Automate and schedule the pipeline's compilation and execution.
2. Set up a daily frequency execution of the workflow you have created.
3. Define release configurations for different environments.
4. Set up workflow configurations to schedule pipeline execution.

Note: If you are stuck and cannot figure out how to proceed after a few minutes, ask the event moderator for help.

🥳🥳Congratulations on completing Lab 3!
You can now move on to Lab 4 for further practice.🥳🥳