Quick tip: Review the prerequisites before you run the lab

End Lab 00:40:23

Caution: When you are in the console, do not deviate from the lab instructions. Doing so may cause your account to be blocked. Learn more.

Open Google Cloud console

Username
student-00-b81be94f2c30(

Password
ifgHju4tHlU9

Project ID
qwiklabs-gcp-01-36b4443l

Create a RAG Application with BigQuery

Lab 1 hour No cost Intermediate

This lab may incorporate AI tools to support your learning.

GSP1289

Google Cloud Self-Paced Labs

Overview

Concerned about AI hallucinations? While AI can be a valuable resource, it sometimes generates inaccurate, outdated, or overly general responses - a phenomenon known as "hallucination." This lab teaches you how to implement a Retrieval Augmented Generation (RAG) pipeline to address this issue. RAG improves large language models (LLMs) like Gemini by grounding their output in contextually relevant information from a specific dataset.

Assume you are helping Coffee-on-Wheels, a pioneering mobile coffee vendor, analyze customer feedback on its services. Without access to the latest data, Gemini's responses might be inaccurate. To solve this problem, you decide to build a RAG pipeline that includes three steps:

1. **Generate embeddings**: Convert customer feedback text into vector embeddings, which are numerical representations of data that capture semantic meaning.
2. **Search vector space**: Create an index of these vectors, search for similar items, and retrieve them.
3. **Generate improved answers**: Augment Gemini with the retrieved information to produce more accurate and relevant responses.

BigQuery allows seamless connection to remote generative AI models on Vertex AI. It also provides various functions for embeddings, vector search, and text generation directly through SQL queries or Python notebooks.

For a deeper dive, check out the course **Create Embeddings, Vector Search, and RAG with BigQuery** on Google Cloud Skills Boost.

What you'll learn

- Create a source connection and grant IAM permissions.
- Generate embeddings and convert text data to vector embeddings.
- Search the vector space and retrieve similar items.
- Generate an improved answer by augmenting Gemini with the search results.

Prerequisites

To complete this lab, you should be familiar with BigQuery and SQL coding.

Setup and requirements

Before you click the Start Lab button

Read these instructions. Labs are timed and you cannot pause them. The timer, which starts when you click **Start Lab**, shows how long Google Cloud resources are made available to you.

This hands-on lab lets you do the lab activities in a real cloud environment, not in a simulation or demo environment. It does so by giving you new, temporary credentials you use to sign in and access Google Cloud for the duration of the lab.

To complete this lab, you need:

- Access to a standard internet browser (Chrome browser recommended).

> **Note:** Use an Incognito (recommended) or private browser window to run this lab. This prevents conflicts between your personal account and the student account, which may cause extra charges incurred to your personal account.

- Time to complete the lab—remember, once you start, you cannot pause a lab.

> **Note:** Use only the student account for this lab. If you use a different Google Cloud account, you may incur charges to that account.

How to start your lab and sign in to the Google Cloud console

1. Click the **Start Lab** button. If you need to pay for the lab, a dialog opens for you to select your payment method. On the left is the Lab Details pane with the following:
    - The Open Google Cloud console button
    - Time remaining
    - The temporary credentials that you must use for this lab
    - Other information, if needed, to step through this lab
2. Click **Open Google Cloud console** (or right-click and select **Open Link in Incognito Window** if you are running the Chrome browser).

    The lab spins up resources, and then opens another tab that shows the Sign in page.

    Tip: Arrange the tabs in separate windows, side-by-side.

    > **Note:** If you see the **Choose an account** dialog, click **Use Another Account**.

3. If necessary, copy the **Username** below and paste it into the **Sign in** dialog.

    ```
    student-00-b81be94f2c30@qwiklabs.net
    ```

    You can also find the Username in the Lab Details pane.

4. Click **Next**.
5. Copy the **Password** below and paste it into the **Welcome** dialog.

    ```
    ifgHju4tHlU9
    ```

    You can also find the Password in the Lab Details pane.

6. Click **Next**.

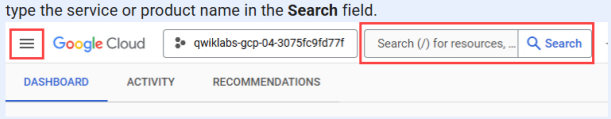
Important: You must use the credentials the lab provides you. Do not use your Google Cloud account credentials.

Note: Using your own Google Cloud account for this lab may incur extra charges.

7. Click through the subsequent pages:
    - Accept the terms and conditions.
    - Do not add recovery options or two-factor authentication (because this is a temporary account).
    - Do not sign up for free trials.

After a few moments, the Google Cloud console opens in this tab.

Note: To access Google Cloud products and services, click the **Navigation menu** or type the service or product name in the **Search** field.



Task 1. Create a source connection and grant IAM permissions

Create a source connection

To use remote generative AI models on Vertex AI in BigQuery, like Gemini and an embedding model, create a new external source connection.

1. In the Google Cloud console, on the **Navigation menu** (☰), click **BigQuery**.
2. Navigate to **Explorer**, click **+ Add**, and select **Connections to external data sources**.

Note: Alternatively, if you do not see the option for **+ Add** followed by **Connections to external data sources**, you can click **+ Add data**, and then use the search bar for data sources to search for **Vertex AI**. Click on the result for **Vertex AI**.

3. In the **Connection type** dropdown, select **Vertex AI remote models, remote functions and BigLake (Cloud Resource)**.
4. In the **Connection ID** field, enter `embedding_conn`.
5. Click **Create connection**.
6. Once the connection is created, click on **Go to connection** in the pop-up confirmation to navigate to the connection and copy the **Service account id** value. You need it later to assign permissions to this account.

Grant IAM permissions

To use BigQuery data and Vertex AI resources, grant the service account the necessary IAM permissions.

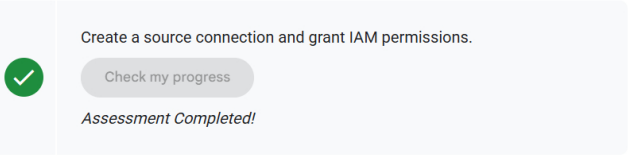
1. Next, you need to grant permissions via IAM. Perform the steps that follow:
    - In the Google Cloud console, on the **Navigation menu** (☰), navigate to **IAM & Admin > IAM**.
    - Click on **Grant access**.
    - In the **Add principals** section:
        - In the **New principals** text field, paste the **Service account id** value

that you copied earlier.

- Under **Assign Role**, select the following roles (search for them if you need to):
  - **BigQuery Data Owner**
  - **Vertex AI User**

2. Click **Save** to apply the changes.
3. Navigate to **APIs and Services** from the **Navigation menu** (☰), click **+ Enable APIs and services**, search `Vertex AI API`, click the **Enable** button.

Click **Check my progress** to verify the objective.



Task 2. Generate embeddings

1. In the Google Cloud console, on the **Navigation menu** (☰), navigate to **BigQuery**.
2. In **Explorer**, navigate to the three dots besides the project, click **Create dataset**. For **Dataset ID**, enter `CustomerReview`. Keep the other option by default, and click **Create dataset**.
3. To connect to the embedding model, run the following SQL query in the query editor:

```
CREATE OR REPLACE MODEL `CustomerReview.Embeddings`
REMOTE WITH CONNECTION `us.embedding_conn`
OPTIONS (ENDPOINT = 'text-embedding-005');
```

4. To upload the dataset from a CSV file, run the following SQL query:

```
LOAD DATA OVERWRITE CustomerReview.customer_reviews
(
    customer_review_id INT64,
    customer_id INT64,
    location_id INT64,
    review_datetime DATETIME,
    review_text STRING,
    social_media_source STRING,
    social_media_handle STRING
)
FROM FILES (
    format = 'CSV',
    uris = ['gs://spls/gsp1249/customer_reviews.csv']
);
```

5. (optional) To check the uploaded data in the table, click **Go to table**. Find the **schema** of the table and **preview** the data.
6. To generate embeddings from recent customer feedback and store them in a table, run the following SQL query in the query editor:

```
CREATE OR REPLACE TABLE
`CustomerReview.customer_reviews_embedded` AS
SELECT *
FROM ML.GENERATE_EMBEDDING(
    MODEL `CustomerReview.Embeddings`,
    (SELECT review_text AS content FROM
`CustomerReview.customer_reviews`)
);
```

7. (Optional) To examine the embedding results, click **Go to table**. Find the **schema**

of the table and **preview** the data. Note that the embedding results are floating-point numbers and may not be immediately interpretable.

Click **Check my progress** to verify the objective.

Generate embeddings.

Check my progress

Assessment Completed!

Task 3. Search the vector space and retrieve the similar items

1. To create an index of the vector search space, run the following SQL query:

Note: For datasets with fewer than 5,000 rows, as in this lab, creating an index is unnecessary. This step demonstrates the code required to create a vector space index when needed for larger datasets.

```
CREATE OR REPLACE VECTOR INDEX `CustomerReview.reviews_index`
ON
`CustomerReview.customer_reviews_embedded`(ml_generate_embedding_re
OPTIONS (distance_type = 'COSINE', index_type = 'IVF');
```

2. To search the vector space and retrieve the similar items, run the following SQL query:

```
CREATE OR REPLACE TABLE `CustomerReview.vector_search_result` AS
SELECT
    query.query,
    base.content
FROM
    VECTOR_SEARCH(
        TABLE `CustomerReview.customer_reviews_embedded`,
        'ml_generate_embedding_result',
        (
            SELECT
                ml_generate_embedding_result,
                content AS query
            FROM
                ML.GENERATE_EMBEDDING(
                    MODEL `CustomerReview.Embeddings`,
                    (SELECT 'service' AS content)
                )
        ),
        top_k => 5,
        options => '{"fraction_lists_to_search": 0.01}'
    );
```

3. (Optional) To check the query results, click **Go to table**. Find the **schema** of the table and **preview** the data.

Click **Check my progress** to verify the objective.

Search the vector space and retrieve the similar items.

Check my progress

Assessment Completed!

Task 4. Generate an improved answer

1. To connect to the Gemini model, run the following SQL query:

```
CREATE OR REPLACE MODEL `CustomerReview.Gemini`
REMOTE WITH CONNECTION `us.embedding_conn`
OPTIONS (ENDPOINT = 'gemini-2.0-flash');
```

2. To enhance Gemini's responses, provide it with relevant and recent data retrieved from the vector search by running the following query:

```
SELECT
    ml_generate_text_llm_result AS generated
FROM
    ML.GENERATE_TEXT(
        MODEL `CustomerReview.Gemini`,
        (
            SELECT
                CONCAT(
                    'Summarize what customers think about our
services',
                    STRING_AGG(FORMAT('review text: %s',
base.content), ',\n')
                ) AS prompt
            FROM
                `CustomerReview.vector_search_result` AS base
        ),
        STRUCT(
            0.4 AS temperature,
            300 AS max_output_tokens,
            0.5 AS top_p,
            5 AS top_k,
            TRUE AS flatten_json_output
        )
    );
```

3. Check the Gemini-generated results in the **Query results** section below the query editor.

Questions for you:
1. How do you determine whether Gemini generates better answers with RAG than without it? Try testing it with code.

2.How can the code be improved? For example, instead of saving vector search results to a table (Task 3), could that process be embedded directly into answer generation (Task 4) for real-time retrieval?

Explore these questions with any remaining lab time. Good luck!

Click **Check my progress** to verify the objective.

Generate the enhanced response with data retrieved from the vector search.

Check my progress

Assessment Completed!

Congratulations!

To help Coffee-on-Wheels gain insights from customer feedback on its services, you successfully implemented a RAG pipeline in BigQuery, providing Gemini with relevant and up-to-date information. You connected to remote generative AI models, including an embedding model and Gemini, and followed three steps: creating embeddings, searching a vector space, and generating an improved answer. The goal is to enable you to apply this same approach to address your own AI hallucination challenges.

Google Cloud training and certification

...helps you make the most of Google Cloud technologies. Our classes include technical skills and best practices to help you get up to speed quickly and continue your learning journey. We offer fundamental to advanced level training, with on-demand, live, and virtual options to suit your busy schedule. Certifications help you validate and prove

your skill and expertise in Google Cloud technologies.

Manual last updated: March 26, 2025

Lab last tested: March 26, 2025

Ready for more?

Here's another lab we think you'll like.

LAB

A Tour of Google Cloud Hands-on Labs

0% Continue