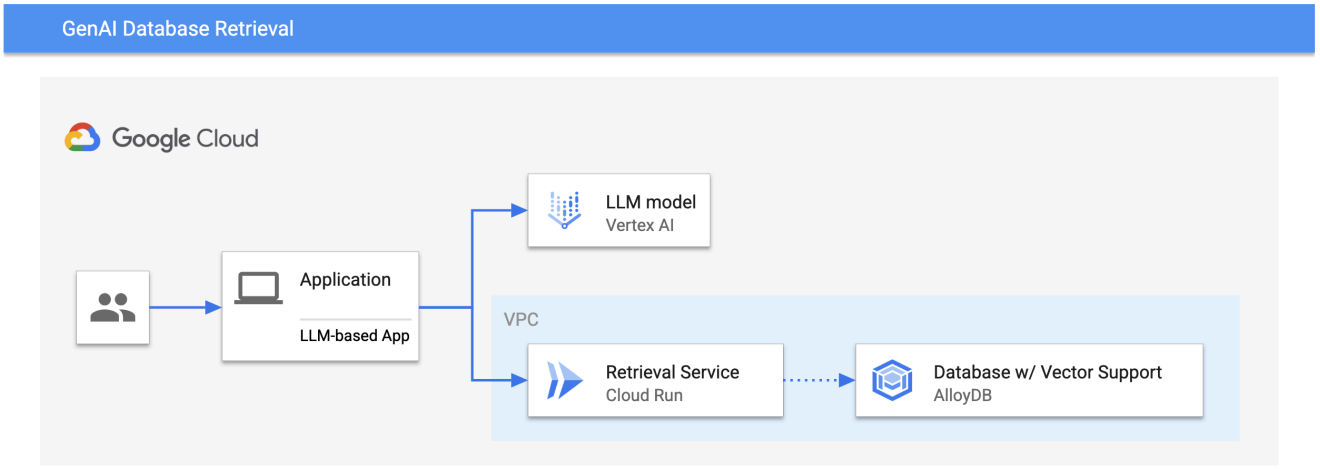
Module 5. Xây dựng ứng dụng chat dựa trên LLM và RAG với AlloyDB và Vertex AI

Tổng quan

Một trong những công cụ tốt nhất để cải thiện chất lượng phản hồi từ các mô hình ngôn ngữ lớn (LLM) là tạo sinh tăng cường truy xuất (RAG). RAG là mô hình truy xuất một số dữ liệu không công khai và sử dụng dữ liệu đó để tăng cường lệnh nhắc (prompt) gửi đến LLM. RAG cho phép LLM tạo ra các phản hồi chính xác hơn dựa trên dữ liệu được bao gồm trong lệnh nhắc.

Bạn sẽ sử dụng AlloyDB, cơ sở dữ liệu tương thích PostgreSQL có khả năng mở rộng và hiệu suất cao của Google Cloud, để lưu trữ và tìm kiếm theo một loại dữ liệu vector đặc biệt gọi là nhúng vector. Nhúng vector có thể được truy xuất bằng tìm kiếm ngữ nghĩa, cho phép truy xuất dữ liệu có sẵn phù hợp nhất với truy vấn ngôn ngữ tự nhiên của người dùng. Dữ liệu được truy xuất sau đó được chuyển đến LLM trong prompt.

Bạn cũng sẽ sử dụng Vertex AI, nền tảng phát triển AI thống nhất, được quản lý hoàn toàn của Google Cloud để xây dựng và sử dụng AI tạo sinh. Ứng dụng của bạn sử dụng Gemini Pro, một mô hình nền tảng đa phương thức hỗ trợ thêm các tệp hình ảnh, âm thanh, video và PDF trong các prompt văn bản hoặc chat và hỗ trợ hiểu ngữ cảnh dài.



Bạn sẽ học được gì

Trong lab này, bạn sẽ học:

- Cách RAG tăng cường khả năng của LLM bằng cách truy xuất thông tin liên quan từ cơ sở kiến thức.
- Cách AlloyDB có thể được sử dụng để tìm thông tin liên quan bằng cách sử dụng tìm kiếm ngữ nghĩa.

- Cách bạn có thể sử dụng Vertex AI và các mô hình nền tảng của Google để cung cấp các khả năng AI tạo sinh mạnh mẽ cho các ứng dụng.

Thiết lập và yêu cầu

Trước khi bạn nhấp vào nút Start Lab

> **Lưu ý: Đọc các hướng dẫn này.**
>
> Các lab được tính thời gian và bạn không thể tạm dừng chúng. Bộ đếm thời gian, bắt đầu khi bạn nhấp vào Start Lab, cho biết thời gian tài nguyên Google Cloud sẽ được cung cấp cho bạn.

Lab thực hành Qwiklabs này cho phép bạn tự thực hiện các hoạt động trong một môi trường đám mây thực, không phải trong môi trường mô phỏng hoặc demo. Điều này được thực hiện bằng cách cung cấp cho bạn thông tin đăng nhập tạm thời mới mà bạn sử dụng để đăng nhập và truy cập Google Cloud trong thời gian diễn ra lab.

Những gì bạn cần

Để hoàn thành lab này, bạn cần:

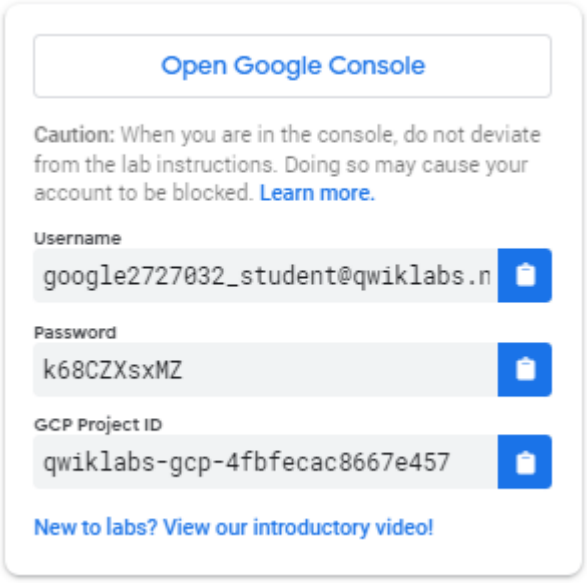
- Truy cập trình duyệt internet tiêu chuẩn (khuyến nghị trình duyệt Chrome).
- Thời gian để hoàn thành lab.

> **Lưu ý:**
> Nếu bạn đã có tài khoản hoặc dự án Google Cloud cá nhân của riêng mình, đừng sử dụng nó cho lab này.

> **Lưu ý:**
>
> Nếu bạn đang sử dụng Pixelbook, hãy mở một cửa sổ ẩn danh để chạy lab này.

Cách bắt đầu lab và đăng nhập vào Console

1. Nhấp vào nút **Start Lab**. Nếu bạn cần thanh toán cho lab, một cửa sổ bật lên sẽ mở ra để bạn chọn phương thức thanh toán. Ở bên trái là một bảng chứa thông tin đăng nhập tạm thời mà bạn phải sử dụng cho lab này.

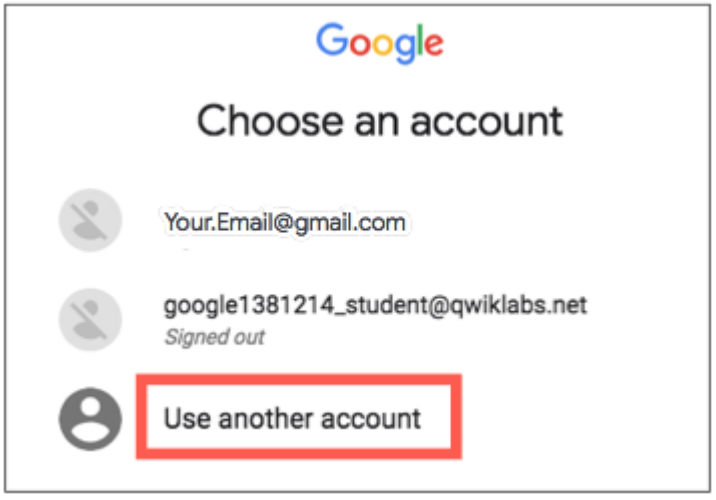


Bảng thông tin đăng nhập

2. Sao chép tên người dùng, sau đó nhấp vào **Open Google Console**. Lab sẽ khởi tạo tài nguyên và sau đó mở một tab khác hiển thị trang **Choose an account**.

> **Lưu ý:**
>
> Mở các tab trong các cửa sổ riêng biệt, cạnh nhau.

3. Trên trang **Choose an account**, nhấp vào **Use Another Account**. Trang **Sign in** sẽ mở ra.



Hộp thoại Choose an account với tùy chọn Use Another Account được làm nổi bật

4. Dán tên người dùng mà bạn đã sao chép từ bảng **Connection Details**. Sau đó sao chép và dán mật khẩu.

> **Lưu ý:**
>
> Bạn phải sử dụng thông tin đăng nhập từ bảng Connection Details. Không sử dụng thông tin đăng nhập Google Cloud Skills Boost của bạn. Nếu bạn có tài khoản Google

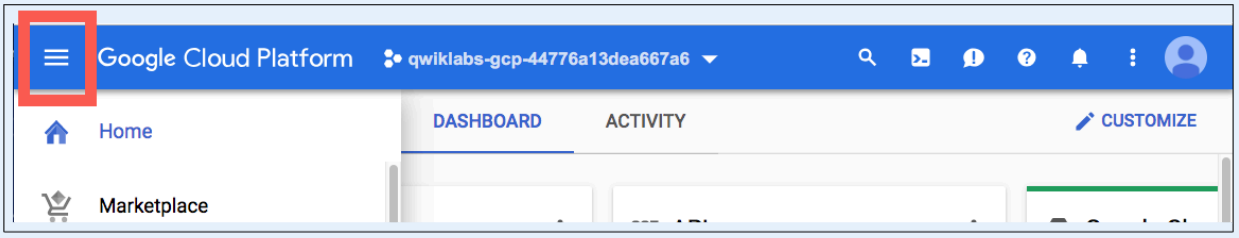
Cloud của riêng mình, đừng sử dụng nó cho lab này (tránh phát sinh phí).

5. Nhấp qua các trang tiếp theo:
   - Chấp nhận các điều khoản và điều kiện.
   - Không thêm các tùy chọn khôi phục hoặc xác thực hai yếu tố (vì đây là tài khoản tạm thời).
   - Không đăng ký các gói dùng thử miễn phí.

Sau vài giây, Cloud console sẽ mở ra trong tab này.

Lưu ý:

Bạn có thể xem menu danh sách các Sản phẩm và Dịch vụ của Google Cloud bằng cách nhấp vào menu Điều hướng ở phía trên bên trái. Menu Cloud Console

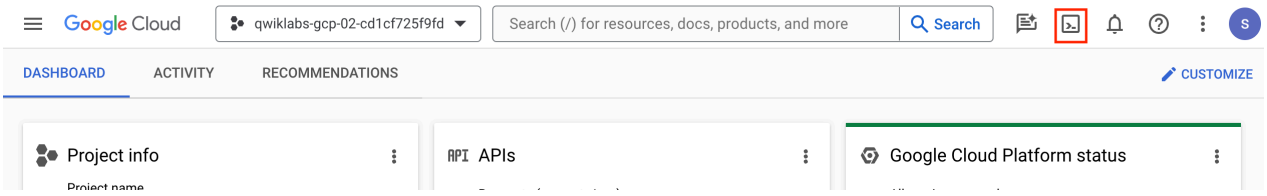


Kích hoạt Google Cloud Shell

Google Cloud Shell là một máy ảo được tải sẵn các công cụ phát triển. Nó cung cấp một thư mục chính 5GB liên tục và chạy trên Google Cloud.

Google Cloud Shell cung cấp quyền truy cập dòng lệnh vào các tài nguyên Google Cloud của bạn.

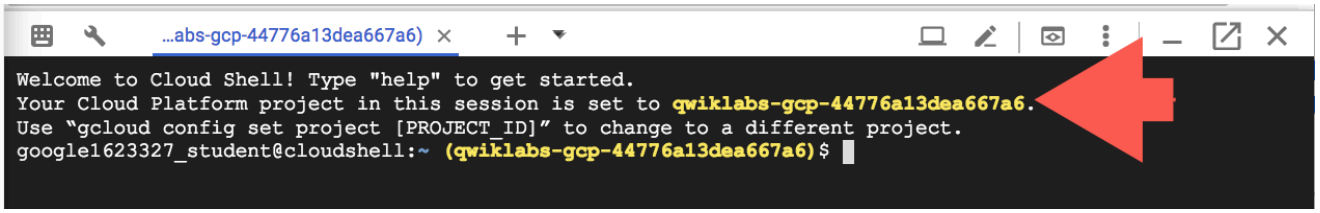
1. Trong Cloud console, trên thanh công cụ phía trên bên phải, nhấp vào nút **Open Cloud Shell**.



Biểu tượng Cloud Shell được làm nổi bật

2. Nhấp vào **Continue**.

Mất vài giây để cấp phát và kết nối với môi trường. Khi bạn đã kết nối, bạn đã được xác thực và dự án được đặt thành PROJECT_ID của bạn. Ví dụ:



ID dự án được làm nổi bật trong Terminal Cloud Shell

`gcloud` là công cụ dòng lệnh cho Google Cloud. Nó được cài đặt sẵn trên Cloud Shell và hỗ trợ tự động hoàn thành bằng phím tab.

Bạn có thể liệt kê tên tài khoản đang hoạt động bằng lệnh này:

```
gcloud auth list
```

Đầu ra:

```
Credentialed accounts:
 - @.com (active)
```

Ví dụ đầu ra:

```
Credentialed accounts:
 - google1623327_student@qwiklabs.net
```

Bạn có thể liệt kê ID dự án bằng lệnh này:

```
gcloud config list project
```

Đầu ra:

```
[core]
project =
```

Ví dụ đầu ra:

```
[core]
project = qwiklabs-gcp-44776a13dea667a6
```

Lưu ý: Tài liệu đầy đủ về gcloud có sẵn trong Hướng dẫn tổng quan về gcloud CLI.
https://www.cloudskillsboost.google/focuses/39961255?parent=lti_session#:~:text=gcloud%20CLI%20overview%20guide

Bài 1. Khởi tạo môi trường cơ sở dữ liệu

Trong bài này, bạn sẽ cài đặt một client PostgreSQL và kết nối nó với instance AlloyDB.

Cài đặt client PostgreSQL

Một máy ảo (VM) đã được tạo. VM này lưu trữ ứng dụng. Bạn cũng tạo client PostgreSQL trên VM này.

1. Để kết nối với VM, hãy chạy lệnh sau:

   ```
   gcloud compute ssh app-vm --zone=ZONE
   ```

   Nếu được yêu cầu ủy quyền, hãy nhấp vào **Authorize**.
2. Đối với mỗi câu hỏi được đưa ra bởi lệnh `gcloud compute ssh`, hãy nhấp vào **Enter** hoặc **Return** để chỉ định đầu vào mặc định.

Sau một thời gian ngắn chờ đợi, bạn sẽ đăng nhập vào VM.

3. Để cài đặt client PostgreSQL, trong phiên VM, hãy chạy các lệnh sau:

   ```
   sudo apt-get update
   sudo apt-get install --yes postgresql-client
   ```

   Lưu ý: Client có thể đã được cài đặt.

Kết nối với instance AlloyDB

Một instance AlloyDB đã được tạo.

1. Để tạo các biến shell cần thiết, hãy chạy lệnh sau:

   ```
   export PGUSER=PG_USER
   export PGPASSWORD=PG_PASSWORD
   export PROJECT_ID=$(gcloud config get-value project)
   export REGION=REGION
   export ADBCLUSTER=CLUSTER
   export INSTANCE_IP=$(gcloud alloydb instances describe $ADBCLUSTER-pr
   --cluster=$ADBCLUSTER --region=$REGION --format="value(ipAddress)")
   ```

2. Để kết nối với instance AlloyDB bằng psql, hãy chạy lệnh sau:

   ```
   psql "host=$INSTANCE_IP user=$PGUSER sslmode=require"
   ```

psql kết nối với cơ sở dữ liệu AlloyDB và hiển thị dấu nhắc `postgres=>`. Bây giờ bạn đã kết nối với cơ sở dữ liệu.

3. Để thoát phiên psql, hãy chạy lệnh sau:

```
exit
```

Lưu ý: Không đóng phiên SSH.

Bài 2. Tạo cơ sở dữ liệu vector

Trong bài này, bạn sử dụng client PostgreSQL để tạo cơ sở dữ liệu AlloyDB và bật nhúng vector.

Tạo cơ sở dữ liệu

1. Để tạo một cơ sở dữ liệu mới, trong phiên VM, hãy chạy lệnh sau:

```
export PGPASSWORD=PG_PASSWORD
psql "host=$INSTANCE_IP user=$PGUSER" -c "CREATE DATABASE
assistantdemo"
```

psql phản hồi với `CREATE DATABASE`.

2. Để cho phép cơ sở dữ liệu hỗ trợ tìm kiếm ngữ nghĩa, các thực thể phải được biểu diễn bằng nhúng vector.
3. Để bật nhúng vector trong cơ sở dữ liệu này, hãy chạy lệnh sau:

```
psql "host=$INSTANCE_IP user=$PGUSER dbname=assistantdemo" -c "CREATE
EXTENSION vector"
```

psql phản hồi với `CREATE EXTENSION`.

4. Nhấp vào **Check my progress** để xác minh mục tiêu.

Tạo cơ sở dữ liệu AlloyDB và bật tiện ích mở rộng pgVector.

Bài 3. Cài đặt Python

Trong bài này, bạn cài đặt Python trong VM. Python được sử dụng để điền dữ liệu vào cơ sở dữ liệu.

1. Để cài đặt Python và Git, trong VM, hãy chạy các lệnh sau:

```
sudo apt install -y python3.11-venv git
python3 -m venv .venv
```

```
source ~/.venv/bin/activate
pip install --upgrade pip
```

Khi quá trình cài đặt hoàn tất, bạn sẽ ở trong môi trường Python ảo, với dấu nhắc `(.venv)`.

2. Nếu phiên SSH của VM bị hết thời gian chờ hoặc tab bị đóng, bạn có thể SSH vào VM lại và sử dụng lệnh `source ~/.venv/bin/activate` để khởi động lại môi trường Python ảo.
3. Để xác nhận phiên bản Python, hãy chạy lệnh sau:

```
python -V
```

Phản hồi của bạn sẽ tương tự như sau:

```
(.venv) student@app-vm:~$ python -V
Python 3.11.2
(.venv) student@app-vm:~$
```

Bài 4. Điền dữ liệu vào cơ sở dữ liệu mẫu

Trong bài này, bạn sẽ điền dữ liệu mẫu vào cơ sở dữ liệu vector trong AlloyDB. Dữ liệu này được sử dụng cho ứng dụng chat mẫu.

Ứng dụng mẫu và dữ liệu được lưu trữ trong một kho lưu trữ GitHub có tên `genai-databases-retrieval-app`.

1. Để sao chép kho lưu trữ, trong VM, hãy chạy lệnh sau:

```
git clone https://github.com/GoogleCloudPlatform/genai-databases-retrieval-app.git
```

2. Để xem mô hình dữ liệu, hãy chạy lệnh sau:

```
cd ~/genai-databases-retrieval-app
cat retrieval_service/models/models.py
```

Các mô hình dữ liệu Python được hiển thị ở đây. Mô hình bao gồm sân bay, chuyến bay, tiện nghi trong nhà ga, chính sách và vé.

3. Để xem ví dụ về dữ liệu sân bay, hãy chạy các lệnh sau:

```
head -1 data/airport_dataset.csv; grep SFO data/airport_dataset.csv
```

Các lệnh này hiển thị tiêu đề CSV chỉ định tên cột cho tập dữ liệu sân bay, tiếp theo là hàng cho sân bay quốc tế San Francisco (SFO). Dữ liệu trong mô hình sân bay có thể được truy xuất dựa trên mã của Hiệp hội Vận tải Hàng không Quốc tế (IATA) hoặc theo quốc gia, thành phố và tên sân bay. Bạn có thể sử dụng tìm kiếm từ khóa để tìm các hàng trong bảng này, vì vậy không có nhúng vector cho dữ liệu này.

4. Để xem ví dụ về dữ liệu chuyến bay, hãy chạy các lệnh sau:

```
head -1 data/flights_dataset.csv; grep -m10 "SFO"
data/flights_dataset.csv
```

Các lệnh này hiển thị tiêu đề CSV chỉ định tên cột cho tập dữ liệu chuyến bay, tiếp theo là 10 hàng chuyến bay đầu tiên đến hoặc đi từ SFO. Dữ liệu trong mô hình chuyến bay có thể được truy xuất dựa trên hãng hàng không và số hiệu chuyến bay hoặc theo mã sân bay đi và đến.

5. Để xem ví dụ về dữ liệu tiện nghi, hãy chạy lệnh sau:

```
head -2 data/amenity_dataset.csv
```

Lệnh này hiển thị tiêu đề CSV chỉ định tên cột cho tập dữ liệu tiện nghi, tiếp theo là tiện nghi đầu tiên.

Bạn sẽ nhận thấy rằng tiện nghi đầu tiên có một số giá trị đơn giản, bao gồm tên, mô tả, vị trí, nhà ga, danh mục và giờ làm việc. Giá trị tiếp theo là nội dung, kết hợp tên, mô tả và vị trí. Giá trị cuối cùng là nhúng, nhúng vector cho hàng.

Nhúng là một mảng gồm 768 số được sử dụng khi thực hiện tìm kiếm ngữ nghĩa. Các nhúng này được tính toán bằng cách sử dụng mô hình AI do Vertex AI cung cấp. Khi người dùng cung cấp một truy vấn, một nhúng vector có thể được tạo từ truy vấn và dữ liệu có nhúng vector gần với nhúng của tìm kiếm có thể được truy xuất.

Dữ liệu chính sách cũng sử dụng nhúng vector theo cách tương tự.

Lưu ý: Việc tính toán nhúng mất một chút thời gian, vì vậy các nhúng đã được cung cấp sẵn. Tập lệnh `run_generate_embeddings.py` có thể được kiểm tra để xem cách tạo nhúng.

6. Để tạo tệp cấu hình cơ sở dữ liệu, hãy chạy các lệnh sau:

```
export PGUSER=PG_USER
export PGPASSWORD=PG_PASSWORD
export PROJECT_ID=$(gcloud config get-value project)
export REGION=REGION
export ADBCLUSTER=CLUSTER
export INSTANCE_IP=$(gcloud alloydb instances describe $ADBCLUSTER-pr
```

```
--cluster=$ADBCLUSTER --region=$REGION --format="value(ipAddress)")
cd ~/genai-databases-retrieval-app/retrieval_service
cp example-config.yml config.yml
sed -i s/127.0.0.1/$INSTANCE_IP/g config.yml
sed -i s/my-user/$PGUSER/g config.yml
sed -i s/my-password/$PGPASSWORD/g config.yml
sed -i s/my_database/assistantdemo/g config.yml
cat config.yml
```

Tệp cấu hình `config.yml` được tạo với địa chỉ IP instance, tên người dùng, mật khẩu và cơ sở dữ liệu được cập nhật. Tệp cấu hình của bạn bây giờ sẽ giống như sau:

```
host: 0.0.0.0
# port: 8080
datastore:
# Example for AlloyDB
kind: "postgres"
host: 10.65.0.2
# port: 5432
database: "assistantdemo"
user: "postgres"
password: "samplepassword"
```

7. Để điền dữ liệu mẫu vào cơ sở dữ liệu, hãy chạy các lệnh sau:

```
pip install -r requirements.txt
python run_database_init.py
```

Lệnh đầu tiên thêm tất cả các gói cần thiết vào môi trường Python ảo và lệnh thứ hai điền dữ liệu vào cơ sở dữ liệu.

Bài 5. Tạo tài khoản dịch vụ cho dịch vụ truy xuất

Trong bài này, bạn tạo một tài khoản dịch vụ cho dịch vụ truy xuất.

Dịch vụ truy xuất chịu trách nhiệm trích xuất thông tin liên quan từ cơ sở dữ liệu. Nó trích xuất thông tin cần thiết từ cơ sở dữ liệu dựa trên yêu cầu từ một ứng dụng AI. Tài khoản dịch vụ này được sử dụng làm danh tính của dịch vụ Cloud Run đó.

Tạo tài khoản dịch vụ

Người dùng SSH không có quyền cho instance dự án để cung cấp cho tài khoản dịch vụ vai trò chính xác. Bạn tạo tài khoản dịch vụ bằng một tab Cloud Shell mới.

1. Trong Cloud Shell, để mở một tab Cloud Shell mới, hãy nhấp vào **Open a new tab (+)**.
2. Để tạo một tài khoản dịch vụ và cấp cho nó các đặc quyền cần thiết, trong tab mới, hãy chạy các lệnh sau:

```
export PROJECT_ID=$(gcloud config get-value project)
gcloud iam service-accounts create retrieval-identity
gcloud projects add-iam-policy-binding $PROJECT_ID \
--member="serviceAccount:retrieval-
identity@$PROJECT_ID.iam.gserviceaccount.com" \
--role="roles/aiplatform.user"
```

   Tài khoản dịch vụ này được cấp vai trò `roles/aiplatform.user`, cho phép dịch vụ gọi Vertex AI.
3. Để đóng tab mới, hãy chạy lệnh sau:
   `exit`
   Tạo tài khoản dịch vụ retrieval-identity.

Bài 6. Triển khai dịch vụ truy xuất lên Cloud Run

Trong bài này, bạn triển khai dịch vụ truy xuất lên Cloud Run.

1. Để triển khai dịch vụ truy xuất, trong tab Cloud Shell SSH của VM, hãy chạy các lệnh sau:

```
export REGION=REGION
cd ~/genai-databases-retrieval-app
gcloud alpha run deploy retrieval-service \
    --source=./retrieval_service/\
    --no-allow-unauthenticated \
    --service-account retrieval-identity \
    --region $REGION \
    --network=default \
    --quiet
```

   Chờ vài phút cho đến khi quá trình triển khai hoàn tất.
2. Để xác minh dịch vụ, hãy chạy lệnh sau:

```
curl -H "Authorization: Bearer $(gcloud auth print-identity-token)"
$(gcloud run services list --filter="(retrieval-service)" --
format="value(URL)")
```

   Nếu bạn thấy thông báo "Hello World", dịch vụ đang hoạt động và phục vụ các yêu cầu.

Triển khai dịch vụ truy xuất.

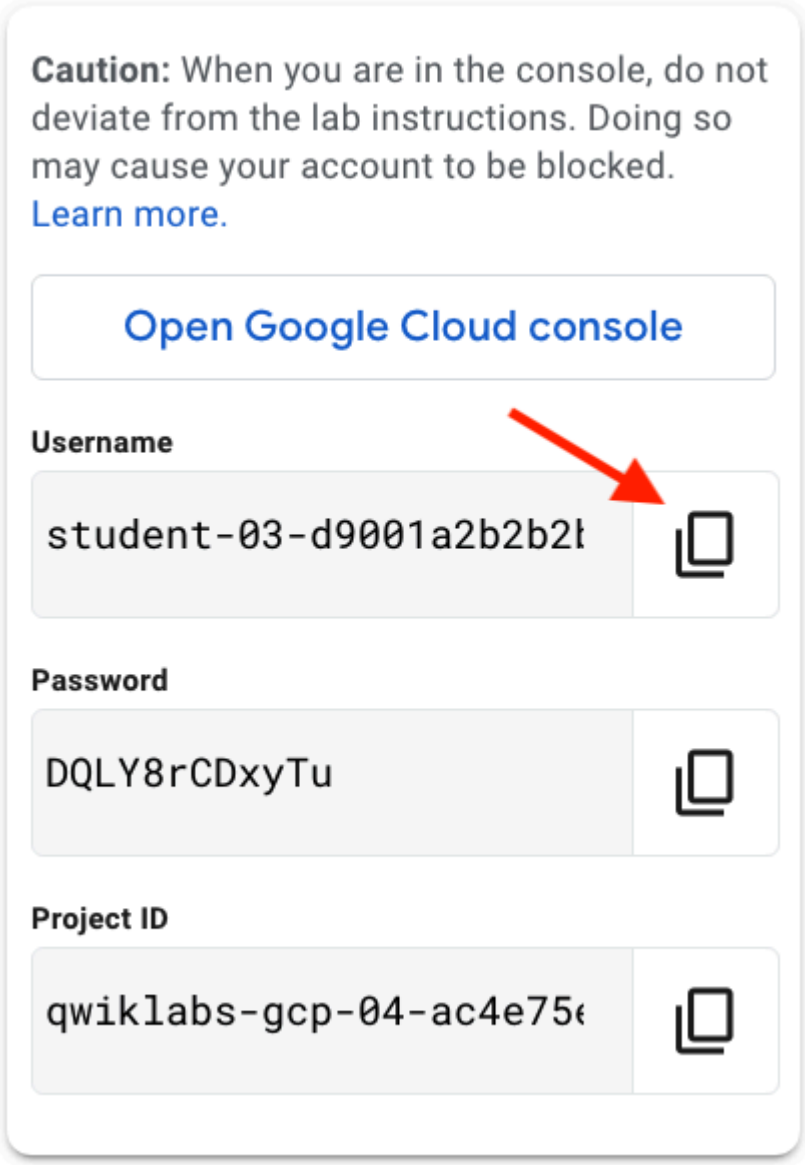
Bài 7. Đăng ký màn hình chấp thuận OAuth

Trong bài này, bạn đăng ký màn hình chấp thuận OAuth được hiển thị cho người dùng đang đăng nhập.

Khi bạn sử dụng OAuth 2.0 để ủy quyền, Google sẽ hiển thị màn hình chấp thuận để thu thập sự đồng ý của người dùng chia sẻ dữ liệu với ứng dụng.

1. Trong Google Cloud console, chọn menu Điều hướng (biểu tượng menu Điều hướng), sau đó chọn **APIs & Services > OAuth consent screen**.
2. Đối với **User Type**, chọn **Internal**, sau đó nhấp vào **Create**.

   Người dùng có quyền truy cập vào dự án sẽ có thể đăng nhập vào ứng dụng.
3. Đối với **App name**, nhập `Cymbal Air`.
4. Nhấp vào **User support email**, sau đó nhấp vào email của sinh viên.
5. Nhấp vào **+ Add Domain**.
6. Đối với **Authorized domain 1**, nhập `cloudshell.dev`.
7. Trên bảng điều khiển bên trái của hướng dẫn lab, sao chép **Username**.



Sao chép username

8. Đối với **Developer contact information**, dán username đã sao chép.
9. Nhấp vào **Save and Continue**.

   Scopes được sử dụng để cho phép người dùng chia sẻ dữ liệu của họ với một ứng dụng. Đối với ứng dụng của chúng ta, chúng ta chỉ sử dụng thông tin cơ bản về người dùng đã đăng nhập, vì vậy không cần scopes.
10. Nhấp vào **Save and Continue**.

    Màn hình chấp thuận hiện đã được thiết lập.

Bài 8. Tạo Client ID cho ứng dụng

Trong bài này, bạn tạo Client ID cho ứng dụng.

Ứng dụng yêu cầu ID ứng dụng khách để sử dụng dịch vụ OAuth của Google. Bạn định cấu hình các nguồn gốc được phép đưa ra yêu cầu này và URI chuyển hướng nơi ứng dụng web được chuyển hướng sau khi người dùng đồng ý đăng nhập.

1. Trong Google Cloud console, chọn menu Điều hướng (biểu tượng menu Điều hướng), sau đó chọn **APIs & Services > Credentials**.
2. Nhấp vào **+ Create Credentials**, sau đó nhấp vào **OAuth client ID**.

   ID ứng dụng khách được sử dụng để xác định một ứng dụng duy nhất cho các máy chủ OAuth của Google.
3. Đối với **Application type**, chọn **Web application**.
4. Đối với **Name**, nhập `Cymbal Air`.

   Bạn có thể tạo nguồn gốc JavaScript và URI chuyển hướng bằng Cloud Shell.
5. Trong Cloud Shell, để mở một tab Cloud Shell mới, hãy nhấp vào **Open a new tab (+)**.
6. Để lấy nguồn gốc và URI chuyển hướng, trong tab mới, hãy chạy các lệnh sau:

```
echo "origin:"; echo "https://8080-$WEB_HOST"; echo "redirect:"; echo "https://8080-$WEB_HOST/login/google"
```

7. Đối với **Authorized JavaScript origins**, nhấp vào **+ Add URI**.

   *Lưu ý: Chọn nút **Add URI** bên dưới **Authorized Javascript origins**, không phải bên dưới **Authorized redirect URIs**.*
8. Sao chép URI nguồn gốc được tạo bởi lệnh `echo` và sau đó, đối với **URIs 1**, dán URI vào đó.
9. Đối với **Authorized redirect URIs**, nhấp vào **+ Add URI**.

   *Lưu ý: Đây là nút **Add URI** thứ hai, bên dưới **Authorized redirect URIs**.*
10. Sao chép URI chuyển hướng được tạo bởi lệnh `echo` và sau đó, đối với **URIs 1**, dán URI vào đó.
11. Nhấp vào **Create**.

    ID ứng dụng khách và secret ứng dụng khách được tạo. Đối với ứng dụng thử nghiệm này, bạn chỉ sử dụng ID ứng dụng khách.
12. Để tạo biến môi trường, trong tab Cloud Shell SSH của VM, hãy dán lệnh sau mà không nhấp vào Enter:

```
export CLIENT_ID=
```

13. Nhấp vào **Copy client ID** (biểu tượng Copy client ID).

    ID ứng dụng khách được sao chép vào clipboard.

    Lưu ý: ID ứng dụng khách cũng có thể được sao chép từ trang Credentials.
14. Trong tab Cloud Shell SSH của VM, dán ID ứng dụng khách, sau đó nhấp vào Enter.

    Lệnh export sẽ tương tự như sau:

```
export CLIENT_ID=937631684809-
q7hs2r191jbks7f7dopih2uafuknb92h.apps.googleusercontent.com
```

Bài 9. Triển khai ứng dụng mẫu

Trong bài này, bạn chạy một ứng dụng chat mẫu sử dụng dịch vụ truy xuất.

Chạy ứng dụng

1. Để cài đặt các yêu cầu Python cho ứng dụng chat, trong tab Cloud Shell SSH của VM, hãy chạy các lệnh sau:

   ```
   cd ~/genai-databases-retrieval-app/llm_demo
   pip install -r requirements.txt
   ```

2. Trước khi khởi động ứng dụng, bạn cần thiết lập một số biến môi trường. Chức năng cơ bản của ứng dụng, bao gồm truy vấn chuyến bay và trả về tiện nghi sân bay, yêu cầu một biến môi trường có tên `BASE_URL` chứa URL cơ sở của dịch vụ truy xuất.
3. Để chỉ định URL cơ sở của dịch vụ truy xuất, hãy chạy các lệnh sau:

   ```
   export BASE_URL=$(gcloud run services list --filter="(retrieval-
   service)" --format="value(URL)")
   echo $BASE_URL
   ```

   URL cơ sở được sử dụng bởi ứng dụng cục bộ để truy cập dịch vụ truy xuất.
4. Để chạy ứng dụng, hãy chạy lệnh sau:

   ```
   python run_app.py
   ```

   Phản hồi của bạn sẽ tương tự như sau:

   ```
   (.venv) student-03-b2f40c6c89d6@app-vm:~/genai-databases-retrieval-
   app/llm_demo$ python run_app.py
   INFO:     Started server process [32894]
   INFO:     Waiting for application startup.
   Loading application...
   INFO:     Application startup complete.
   INFO:     Uvicorn running on http://0.0.0.0:8081 (Press CTRL+C to
   quit)
   ```

   Ứng dụng hiện đang chạy.

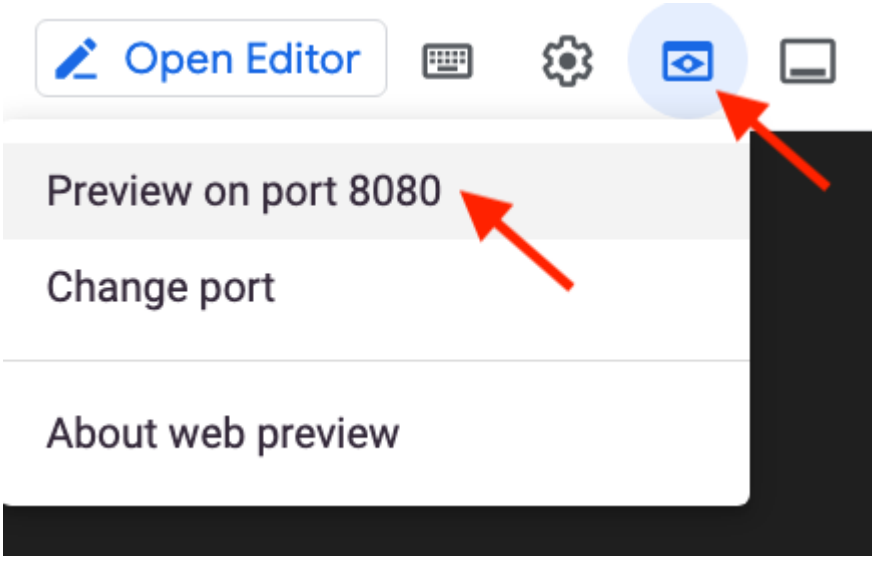
Kết nối với VM

Bạn có một số cách để kết nối với ứng dụng đang chạy trên VM. Ví dụ: bạn có thể mở cổng 8081 trên VM bằng cách sử dụng các quy tắc tường lửa trong VPC hoặc tạo một bộ cân bằng tải với IP công khai. Ở đây, bạn sử dụng một đường hầm SSH đến VM, chuyển cổng 8080 của Cloud Shell sang cổng 8081 của VM.

1. Trong Cloud Shell, để mở một tab Cloud Shell mới, hãy nhấp vào **Open a new tab (+)**.
2. Để tạo một đường hầm SSH đến cổng VM, trong tab mới, hãy chạy lệnh sau:

```
gcloud compute ssh app-vm --zone=ZONE -- -L 8080:localhost:8081
```

   Lệnh `gcloud` kết nối cổng 8080 trong Cloud Shell với cổng 8081 trên VM. Bạn có thể bỏ qua lỗi "Cannot assign requested address."
3. Để chạy ứng dụng trong trình duyệt web, hãy nhấp vào **Web Preview**, sau đó chọn **Preview on port 8080**.



Web Preview trên cổng 8080

```
Một tab mới được mở trong trình duyệt và ứng dụng đang chạy. Ứng dụng
Cymbal Air hiển thị thông báo "Welcome to Cymbal Air! How may I assist
you?"
```

4. Nhập truy vấn sau:

```
When is the next flight to Dallas?
```

   Ứng dụng phản hồi với chuyến bay tiếp theo từ SFO đến Dallas/Fort Worth.
5. Nhập truy vấn sau:

```
Which restaurants are near the departure gate?
```

Ứng dụng hiểu ngữ cảnh và phản hồi với các nhà hàng gần cổng khởi hành ở SFO.

Bài 10. Đăng nhập vào ứng dụng (tùy chọn)

Trong bài này, bạn đăng nhập vào ứng dụng để đặt chuyến bay.

1. Nhấp vào **Sign in**.

   Một cửa sổ bật lên sẽ mở ra.
2. Trong cửa sổ bật lên, chọn sinh viên.

   Tài khoản sinh viên được đăng nhập.
3. Nếu bạn được yêu cầu xác nhận rằng bạn muốn đăng nhập với tư cách sinh viên, hãy nhấp vào **Confirm**.
4. Nhập truy vấn sau:

   ```
   Please book that flight.
   ```

   Ứng dụng hiển thị chuyến bay đang được thảo luận.
5. Nhấp vào **Looks good to me. Book it.**

   Chuyến bay được đặt.
6. Nhập truy vấn sau:

   ```
   Which flights have I booked?
   ```

   Chuyến bay bạn vừa đặt được hiển thị.

Ứng dụng chat có thể giúp trả lời các câu hỏi của người dùng như:

- When is the next flight to Miami?
- Are there any luxury shops around gate D50?
- Where can I get coffee near gate A6?

Ứng dụng sử dụng các mô hình nền tảng mới nhất của Google để tạo ra các phản hồi và tăng cường chúng bằng thông tin về các chuyến bay và tiện nghi từ cơ sở dữ liệu AlloyDB đang hoạt động. Bạn có thể đọc thêm về ứng dụng demo này trên trang GitHub của dự án.